

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-  
GRANDENSE - CÂMPUS PASSO FUNDO  
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

**ISMAEL FELIPE HEPP**

**MNEMOSAPP: Aplicativo da Web Progressivo de auxílio ao  
aprendizado utilizando flashcards**

**Jorge Luis Boeira Bavaresco**

**PASSO FUNDO**

**2021**

**ISMAEL FELIPE HEPP**

**MNEMOSAPP: Aplicativo da Web Progressivo de auxílio ao  
aprendizado utilizando flashcards**

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador (a): Me. Jorge Luis Boeira  
Bavaresco

**PASSO FUNDO**

**2021**

**ISMAEL FELIPE HEPP**

**MNEMOSAPP: Aplicativo da Web Progressivo de auxílio ao aprendizado  
utilizando flashcards**

Trabalho de Conclusão de Curso aprovado em \_\_\_\_/\_\_\_\_/\_\_\_\_ como requisito  
parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

---

Jorge Luis Boeira Bavaresco

---

Josué Toebe

---

Élder Francisco Fontana Bernardi

---

Rafael Marisco Bertei

**PASSO FUNDO**

**2021**

## DEDICATÓRIA

*Dedico às estrelas da minha vida,  
meus pais, minha irmã, família e amigos,  
e à minha lua, Luana,  
que iluminam essa longa noite.*

## **AGRADECIMENTOS**

Em primeiro lugar, gostaria de agradecer aos meus pais, Marta e Pedro, minha irmã, Ethiane, e meu amor, Luana, que sempre me incentivam e apoiam em minhas escolhas, e com quem eu sei que sempre poderei contar.

Aos meus amigos, por estarem presentes, mesmo que longe, por me apoiarem e me ajudarem a controlar a minha excessiva autocrítica.

Ao professor Jorge Luis Boeira Bavaresco, que aceitou me ajudar nessa jornada, me auxiliou no desenvolvimento deste trabalho e que entendeu os momentos de correria e de dificuldade de escrita.

Aos demais professores do IFSul e aos outros servidores, todos que fizeram parte dessa jornada e fazem dessa instituição um marco tão importante na minha vida e de muitos outros.

Por fim, a todos os professores que passaram pela minha vida, desde os primeiros, meus pais, a primeira “oficial”, a dona Ida, e a todos que acreditam que a educação é essencial para se construir uma sociedade melhor, é um direito de todos e é algo pelo qual vale a pena lutar.

## RESUMO

A partir de pesquisas realizadas sobre o ensino a distância, verificou-se que a procura por esta modalidade tem aumentado nos últimos anos. Na modalidade do ensino a distância são utilizados teleaulas, textos digitais, livros eletrônicos, vídeos de outros gêneros e objetos digitais de aprendizagem. Com isto decidiu-se desenvolver um aplicativo que pudesse ser utilizado como ferramenta de ensino a distância, que se enquadra como um objeto digital de aprendizagem. O aplicativo tem como objetivo o auxílio ao aprendizado, baseando-se na técnica de repetição espaçada com o uso de *flashcards*, utilizando o sistema Leitner. Assim, o aplicativo foi desenvolvido usando o *framework* Angular, utilizando a metodologia de desenvolvimento de aplicativo da web progressivo e utilizando os serviços de autenticação de usuário, banco de dados e hospedagem do Firebase. O resultado foi uma aplicação que permite um usuário criar baralhos com qualquer tema e disponibilizá-los para qualquer outro usuário cadastrado para estudos.

Palavras-chave: Flashcard. Repetição espaçada. Angular. PWA. Firebase.

## **ABSTRACT**

From research on distance learning, it was found that the demand for this modality has increased in recent years. In the distance learning modality, on-line classes, digital texts, electronic books, videos of other genres and digital learning objects are used. From this research, it was decided to develop an application that could be used as a distance learning tool, that fits as a digital learning object. The application aims to aid learning, based on the spaced repetition technique using flashcards, using the Leitner system. Thus, the application was developed using the Angular framework, using the progressive web application development methodology and using Firebase's user authentication, database and hosting services. The result was an application that allows a user to create decks with any theme and make them available to any other user registered for studies.

Keywords: Flashcard. Spaced repetition. Angular. PWA. Firebase.

## LISTA DE FIGURAS

Figura 1 - Exemplo de <i>flashcard</i> .....	13
Figura 2 - Curva de esquecimento.....	14
Figura 3 - Sistema Leitner .....	15
Figura 4 - Arquitetura do sistema.....	19
Figura 5 - Diagrama de sequência de fluxo de inscrição.....	22
Figura 6 - Diagrama de atividade de fluxo de estudo.....	24
Figura 7 - Tela inicial do Visual Studio Code .....	25
Figura 8 - Extensões do Visual Studio Code.....	26
Figura 9 - Bibliotecas do projeto.....	28
Figura 10 - Serviços disponibilizados pelo Firebase.....	30
Figura 11 - Tela Inicial do Mnemosapp.....	32
Figura 12 - AppModule .....	33
Figura 13 - Tela Autenticação do Mnemosapp .....	34
Figura 14 - Tela Inicial do Mnemosapp com usuário logado.....	35
Figura 15 - Tela de listagem de baralhos .....	35
Figura 16 - Tela de criação de baralho .....	36
Figura 17 - Tela de criação de baralho - criação de cartas .....	36
Figura 18 - Tela de listagem de baralhos com novo baralho .....	37
Figura 19 - Tela de edição de baralho .....	38
Figura 20 - Tela de edição - baralho compartilhado .....	39
Figura 21 - Tela de criação e edição de carta.....	40
Figura 22 - Tela de listagem de baralhos com baralho inscrito .....	41
Figura 23 - Tela de listagem de baralhos com cópia de baralho inscrito .....	42
Figura 24 - Tela da carta .....	43
Figura 25 - Tela da carta com resposta revelada.....	44
Figura 26 - Fim de rodada.....	45

## LISTA DE QUADROS

Quadro 1 - Requisitos Funcionais .....	20
Quadro 2 - Requisitos não funcionais .....	21

## LISTA DE ABREVIATURAS E SIGLAS

ABED - Associação Brasileira de Educação a Distância

EAD - Ensino a Distância

HTTPS - *Hypertext Transfer Protocol Secure*

IDE - *Integrated Development Environment*

OS - *Operating System*

PWA - *Progressive Web App*

URL - *Uniform Resource Locator*

## SUMÁRIO

1	INTRODUÇÃO.....	10
1.1	OBJETIVOS.....	12
1.1.1	Objetivo geral.....	12
1.1.2	Objetivos específicos.....	12
2	REFERENCIAL TEÓRICO.....	13
2.1	FLASHCARDS.....	13
2.2	REPETIÇÃO ESPAÇADA.....	14
2.3	SISTEMA LEITNER.....	15
2.4	APLICATIVO DA WEB PROGRESSIVO.....	16
2.5	SERVICE WORKER.....	16
3	METODOLOGIA.....	18
3.1	PROPOSTA PARA O SISTEMA.....	18
3.2	ARQUITETURA E MODELAGEM DO SISTEMA.....	19
3.2.1	Requisitos funcionais.....	20
3.2.2	Requisitos não funcionais.....	21
3.2.3	Diagrama de sequência.....	22
3.2.4	Diagrama de atividade.....	23
4	DESENVOLVIMENTO DO APLICATIVO.....	25
4.1	FERRAMENTAS.....	25
4.2	FRAMEWORK.....	27
4.3	BIBLIOTECAS.....	28
4.4	SERVIÇOS UTILIZADOS.....	29
5	RESULTADOS.....	32
6	CONCLUSÃO.....	46
	REFERÊNCIAS.....	47

## 1 INTRODUÇÃO

O tema escolhido para o presente trabalho foi o desenvolvimento de um aplicativo da web progressivo de auxílio ao aprendizado, através da técnica de repetição espaçada e do uso de *flashcards*.

Segundo a Associação Brasileira de Educação a Distância (ABED) a procura pela modalidade de Ensino a Distância (EAD), seja no ensino superior ou em cursos profissionalizantes, tem aumentado. Em 2016, foram contabilizadas 3.734.887 matrículas nessa modalidade. Em 2017, o número aumentou para 7.773.828 matrículas. E, por fim, em 2018, a pesquisa contabilizou 9.374.647 matrículas (ABED, 2019).

Esta pesquisa cita diversos recursos educacionais que são oferecidos aos alunos na modalidade EAD. Entre eles, teleaulas, textos digitais, livros eletrônicos, vídeos de diferentes gêneros que não sejam teleaulas, objetos digitais de aprendizagem, entre outros. O aplicativo proposto se enquadra na modalidade de objetos digitais de aprendizagem que, segundo a pesquisa, encontra-se em quinto lugar dentre os recursos educacionais mais oferecidos aos alunos, sendo utilizado em 69,6% em cursos totalmente a distância (ASSUMPÇÃO, 2019).

Dessa forma, entende-se que o aplicativo proposto, ao utilizar a técnica de repetição espaçada, seria mais uma ferramenta para auxiliar no aprendizado, assim como apontado na pesquisa citada.

Sua diferença, dentre os aplicativos do gênero pesquisados, é seu uso voltado para o ensino a distância, ou seja, terá possibilidade da definição dos conteúdos dos *flashcards* serem elaborados por um professor, para o uso de vários alunos, de acordo com a matéria que esteja sendo estudada.

A escolha de um aplicativo progressivo da web (PWA, *Progressive Web App*, no original em inglês) se deve ao incentivo que a tecnologia recebe de grandes empresas da área, como Google e Microsoft. Segundo Rocha (2020), “as duas gigantes da tecnologia acabam de unir suas forças para trazer mais aplicativos progressivos da web dispositivos *Android* através da *Play Store*”.

No ano de 2020, a Google substituiu aplicativos nativos do Android por versões PWA, em seu sistema operacional, o Chrome OS:

“A Google começou a substituir aplicativos do Android presentes no Chrome OS por versões em Progressive Web Apps. [...] o Twitter foi um dos primeiros serviços a ter seu aplicativo nativo extinto no Chrome OS para adotar o PWA. A Google também está trabalhando para implementar a novidade no YouTube TV. Com a transição, a Play Store identifica quando está rodando em Chromebooks e evita o download da solução nativa para Android, tornando o PWA o novo padrão para os dispositivos.” (MOGNON, 2020).

Já a Microsoft tem testado versões PWA dos seus aplicativos *Office* (Word, Excel, PowerPoint e Outlook), trazendo a instalação por padrão, através de seu navegador Edge (FOLEY, 2020).

Assim chegamos a seguinte questão, é possível auxiliar o aprendizado de um usuário, utilizando a técnica de repetição espaçada com *flashcards*, com um aplicativo da web progressivo? Para resolver este problema, foi feita a modelagem e desenvolvimento de um aplicativo que atendesse às especificações trazidas durante este trabalho.

Para relatar o processo de desenvolvimento do sistema proposto e o referencial teórico em que se baseia, o presente trabalho está dividido da seguinte maneira:

No capítulo 2, “Referencial Teórico”, serão apresentados os conceitos de *flashcard*, de repetição espaçada, do sistema Leitner, de aplicativo da *web* progressivo e de *service worker*, também serão apresentadas as tecnologias utilizadas no desenvolvimento do aplicativo.

No capítulo 3, “Metodologia”, será apresentada a arquitetura e modelagem do sistema, seus requisitos funcionais e não funcionais, e os diagramas de suas principais funcionalidades.

O capítulo 4, “Desenvolvimento do aplicativo”, serão apresentadas as ferramentas de desenvolvimento, bibliotecas, *frameworks* e recursos utilizados na implementação do sistema. Também será explicado como foi feita a integração do aplicativo com o banco de dados, a estrutura dos componentes do aplicativo e como se relacionam e a lógica aplicada no aplicativo para atender aos conceitos que embasam o trabalho.

No capítulo 5, “Resultados”, serão expostos os resultados obtidos ao longo do trabalho desenvolvido, as funcionalidades implementadas e as telas do aplicativo.

E por fim o capítulo 6, Conclusão, onde será realizado o fechamento do trabalho relacionando os objetivos do mesmo com os resultados alcançados ao

longo do trabalho desenvolvido. Ainda, será realizado um levantamento dos pontos que não atenderam os objetivos propostos, para serem incrementados em trabalhos futuros.

Com a estrutura do trabalho definida, a seguir serão citados os objetivos deste trabalho.

## **1.1 OBJETIVOS**

Nesta seção serão apresentados os objetivos gerais e específicos deste trabalho.

### **1.1.1 Objetivo geral**

Aplicar os conceitos relacionados a análise e desenvolvimento de sistemas na web para modelar, desenvolver e implementar um aplicativo da web progressivo de auxílio ao aprendizado, baseada na técnica de repetição espaçada com o uso de *flashcards*.

### **1.1.2 Objetivos específicos**

Com base no objetivo geral, foram definidos os seguintes objetivos específicos:

- Pesquisar sistemas que implementam as técnicas de repetição espaçada.
- Analisar os requisitos específicos ao sistema proposto.
- Especificar as funcionalidades que serão desenvolvidas.
- Apresentar as tecnologias que serão utilizadas no desenvolvimento do sistema proposto.
- Desenvolver o sistema especificado.
- Implantar o sistema desenvolvido.

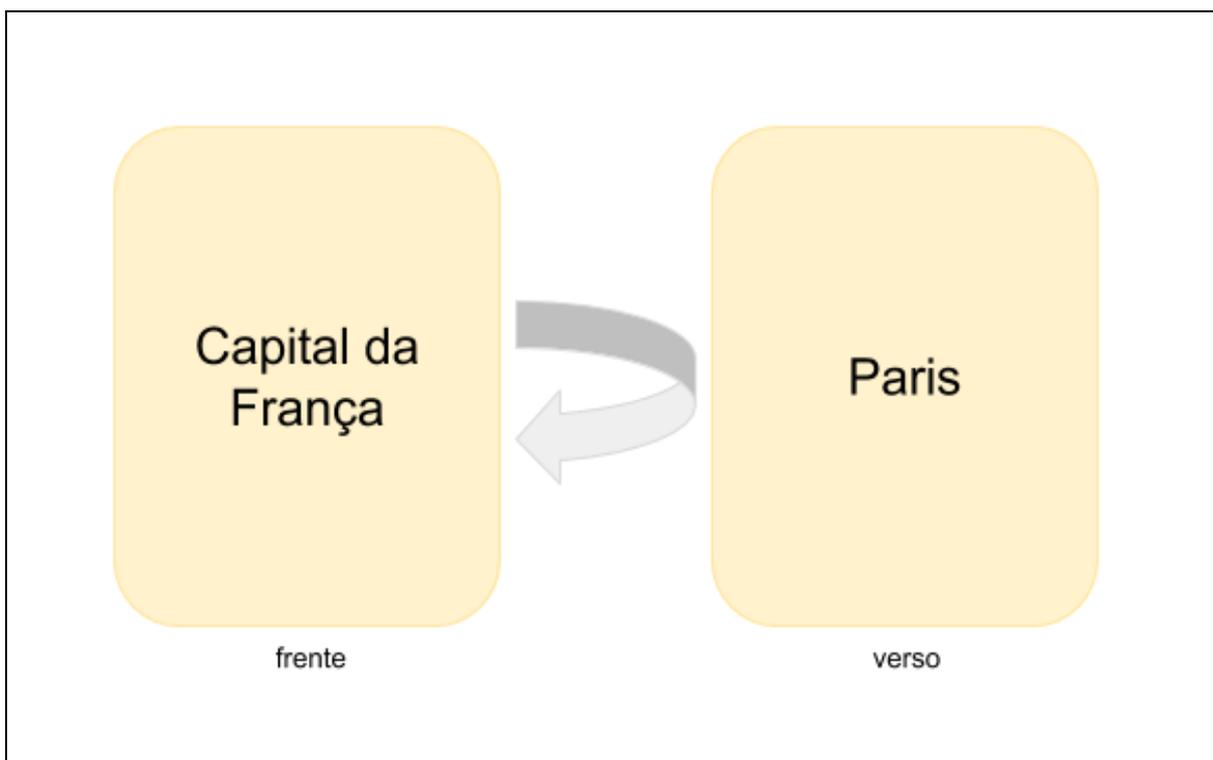
## 2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados os principais conceitos que fundamentam o desenvolvimento do sistema proposto.

### 2.1 FLASHCARDS

*Flashcards* são uma ferramenta de auxílio ao aprendizado. São cartões de papel que podem ser utilizados de duas maneiras: usado para anotação dos tópicos importantes em relação a um conteúdo, ou, como será empregado no sistema proposto, contendo uma pergunta em um lado e sua respectiva resposta no verso (CAMARGO, 2020), conforme exemplificado na “Figura 1 - Exemplo de *flashcard*”.

Figura 1 - Exemplo de *flashcard*



Fonte: Elaborado pelo autor, 2020

Esta “pergunta” pode ser substituída por um termo, equação, figura, vídeo ou áudio (estes dois no caso da implementação dos *flashcards* em um ambiente multimídia), ou seja, qualquer “gatilho” de informação que faça o usuário relacionar o mesmo com o conteúdo que está sendo estudado. Já a “resposta” pode ser desde

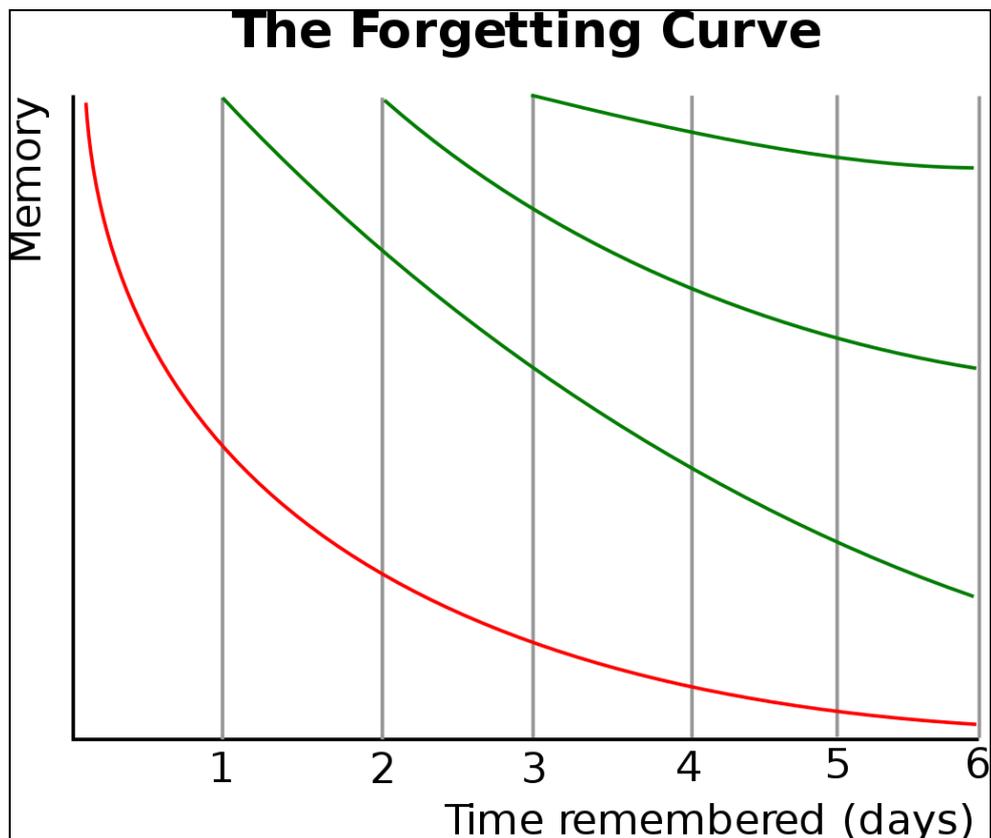
uma resposta direta à pergunta feita, como a solução de uma equação, tradução de um termo, explicação de um tópico, ou seja, a informação a ser aprendida.

O usuário, durante seus estudos, lê a frente de cada cartão e, após lembrar o conteúdo relacionado ou desistir, vira o cartão para conferir se lembrou corretamente ou relembrar a informação já esquecida. Desta forma, esta ferramenta auxilia no aprendizado, através da memorização de um conteúdo.

## 2.2 REPETIÇÃO ESPAÇADA

A repetição espaçada é uma técnica que propõe a revisão de um conteúdo repetidas vezes, variando o intervalo de tempo entre cada uma das revisões. Seu objetivo é a transferência das informações da memória imediata para a memória de longo prazo, o que ocorre pelas revisões periódicas e espaçadas (AZEVEDO, 2017).

Figura 2 - Curva de esquecimento



Fonte: CURVA DO ESQUECIMENTO, 2020

Segundo Israel e North (2014, apud AZEVEDO, 2017, p.18), “são necessárias de 5 a 6 repetições para que uma informação seja incorporada na memória de longo

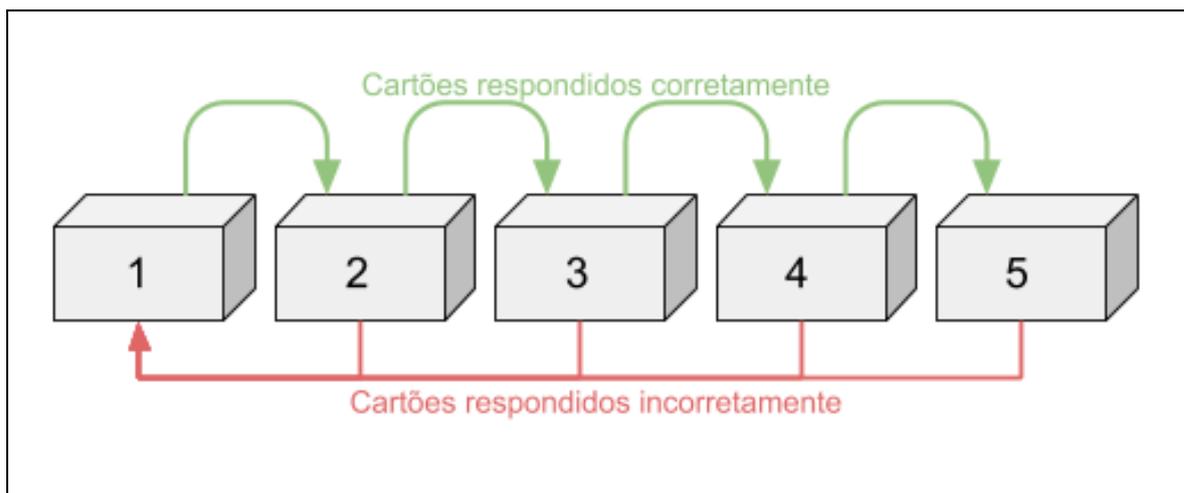
prazo”, conforme exemplo da “Figura 2 - Curva de esquecimento”, que ilustra o período em que uma informação permanece na memória em comparação ao número de vezes em que a informação é revista.

### 2.3 SISTEMA LEITNER

O Sistema Leitner é baseado na técnica de repetição espaçada, utilizando os *flashcards* como ferramenta para sua implementação. Foi criado na década de 1970, pelo jornalista científico alemão Sebastian Leitner, e consiste em um sistema de agendamento de *flashcards*. Ele categoriza os *flashcards* em diferentes blocos, conforme a avaliação do usuário. Se o usuário acertar a pergunta do *flashcard*, este é movido para um bloco que será revisto com uma frequência menor. Caso contrário, o usuário erre, o *flashcard* é movido para um bloco onde será revisado com uma frequência maior (SILVA, 2015).

Tal lógica pode ser visualizada conforme ilustrado na “Figura 3 - Sistema Leitner”. Onde cada bloco representa a categoria onde um *flashcard* se encontra e o número a sua classificação, que determina a frequência de exibição, sendo 1 a maior frequência e 5 a menor.

Figura 3 - Sistema Leitner



Fonte: Elaborado pelo autor, 2020

## 2.4 APLICATIVO DA WEB PROGRESSIVO

Um PWA é um aplicativo web desenvolvido seguindo uma metodologia de desenvolvimento que, através do uso de recursos existentes nos *browsers*, se utiliza de tecnologias da *web* (como HTML, CSS, Javascript) e de recursos disponíveis aos aplicativos nativos.

Seu conceito foi definido por Alex Russel e Frances Berriman, que enumeram as características necessárias para que um aplicativo seja considerado um aplicativo *web* progressivo. Segundo Russel (2015), elas são:

- Responsivo: deve se adequar a qualquer dimensão de tela;
- Independente de conectividade: Progressivamente aprimorado com *service workers* para executar *off-line*;
- Semelhante a um *app*: apresentar interações e navegação semelhantes a um aplicativo nativo;
- Atualizado: Sempre atualizado através do processo de atualização do *service worker*;
- Seguro: A conexão deve ser feita através do protocolo HTTPS;
- Descobrível: São identificados através de manifestos e do escopo de registro dos *service workers*, possibilitando que sejam encontrados por mecanismos de busca da web;
- Reengajável: Possuem acesso aos recursos de reengajamento do sistema, como, por exemplo, notificações *push*;
- Instalável: Através do *browser* é possível adicionar o aplicativo à tela inicial do sistema, sem precisar acessar uma loja de aplicativos;
- Linkável: Pode ser compartilhado através de uma URL, sem necessidade de uma instalação complexa.

## 2.5 SERVICE WORKER

O *service worker* é uma categoria de *web worker*, ou seja, é um *script* JavaScript que executa em segundo plano, independentemente dos *scripts* que estejam em execução na página principal (WHATWG, 2020). O que o diferencia de outros *web workers* é a sua função específica, ele é um *proxy* de rede programável,

ou seja, “[...] permite controlar como as solicitações de rede da página são gerenciadas” (GAUNT, 2020). Além disso, segundo o autor, possui recursos como notificações *push* e sincronização em segundo plano, permitindo um maior controle da experiência do usuário e possibilitando o desenvolvimento de aplicativos *web* que funcionem mesmo estando *off-line*.

### 3 METODOLOGIA

Neste capítulo será apresentada a proposta e funcionalidades do sistema, sua arquitetura e modelagem, os requisitos funcionais e não funcionais, e os diagramas de suas principais funcionalidades.

#### 3.1 PROPOSTA PARA O SISTEMA

O sistema proposto neste trabalho se diferencia por se tratar de uma plataforma para a aplicação da técnica dos *flashcards* que será implementada como um aplicativo da *web* progressivo. Conseqüentemente, não estará atrelado a um sistema específico para ser executado e após a instalação poderá operar com uma conexão com a internet de baixa velocidade, ou mesmo sem conexão à internet. A seguir será apresentado o funcionamento do sistema proposto.

Um usuário, após seu registro, poderá criar um baralho. Neste processo de criação deve, juntamente, criar um número mínimo de cartas para este baralho, no momento foi definido que este número inicial será de três cartas. Após criado, o baralho estará disponível para ser jogado em uma rodada de estudos, editado, deletado ou compartilhado.

Ao iniciar uma rodada de estudos nova com um baralho o sistema cria uma classificação específica do usuário com aquele baralho. Isto separa cada carta em um "bloco" que determinará a frequência com que ela será sorteada para exibição.

Na rodada de estudos é sorteada uma carta com base na classificação de todas as cartas do baralho, que inicialmente, ao ser criada, é uma classificação padrão. A carta sorteada é apresentada ao usuário, que, após ponderar sobre a pergunta nela retratada, pode solicitar ao sistema que exiba a resposta dela. Com a resposta, o usuário pode marcar aquela rodada como correta ou errada, dependendo se conseguiu ou não lembrar da resposta corretamente. Caso marque como incorreta, a carta é reclassificada para um bloco inferior, onde a o "peso" da carta aumenta, ou seja, em futuras rodadas a carta terá mais chances de ser sorteada. Do contrário, caso marque como correta, o oposto ocorre, a carta é reclassificada para um bloco superior, onde seu "peso" diminui, reduzindo a frequência com que ela será sorteada em futuras rodadas.

Ainda é possível criar novas cartas para um baralho existente, editar as cartas existentes ou deletá-las. Todas estas operações levam em conta o compartilhamento de baralho, onde as alterações feitas no baralho, ou nas cartas, refletem nos usuários inscritos no baralho alterado assim que seus sistemas se conectarem à internet. A classificação das cartas também é atualizada conforme ocorrem alterações nas cartas de um baralho.

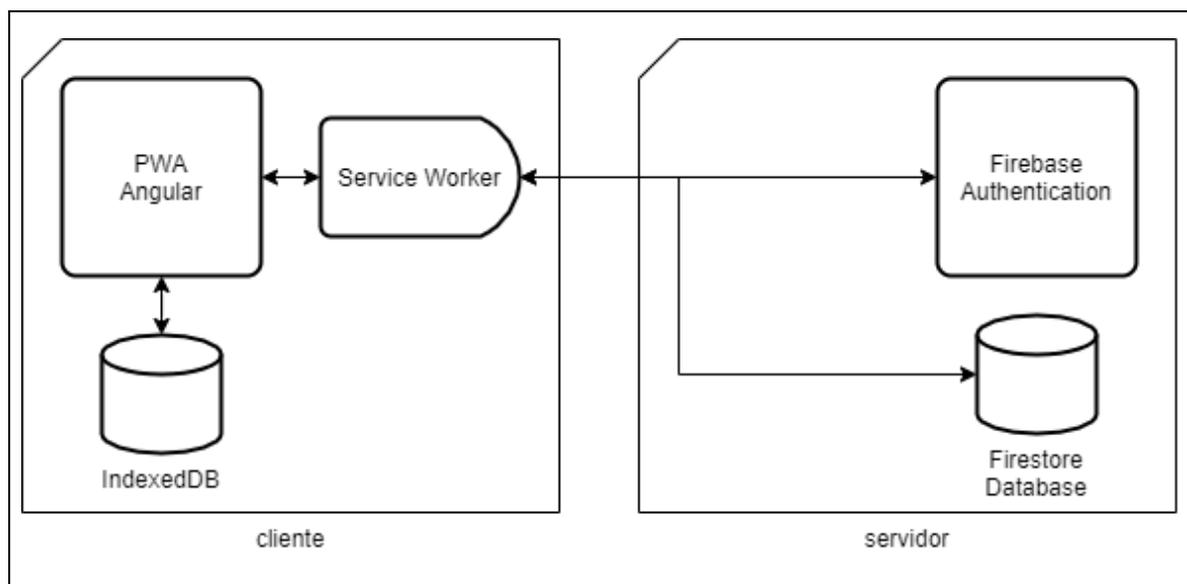
Do ponto de vista da implementação das regras do sistema, o compartilhamento dos baralhos criados é simples, bastando acessar um *link*, e os temas dos baralhos são livres, podendo ser utilizados em qualquer área. Por fim, será um sistema nativamente em português.

A fácil instalação, compartilhamento simples de baralho e para vários usuários, não restringir no tema dos baralhos e ser um aplicativo totalmente em português, são características que tornam o aplicativo mais inclusivo, que diferenciam o Mnemosapp das alternativas existentes.

### 3.2 ARQUITETURA E MODELAGEM DO SISTEMA

O sistema possui uma arquitetura cliente-servidor, conforme a “Figura 4 - Arquitetura do sistema”.

Figura 4 - Arquitetura do sistema



Fonte: Elaborado pelo autor, 2021

Do lado do servidor foram utilizados os serviços do *Firebase* da Google, em específico o serviço de autenticação, *Firebase Authentication*, o serviço de banco de dados não relacionais, *Cloud Firestore*, e o *Firebase Hosting*, serviço que hospeda o aplicativo. Além disso, ele servirá a PWA, que poderá ser instalada no dispositivo do usuário.

O cliente foi desenvolvido com o *framework* Angular, implementando as características necessárias para ser considerado um PWA. Por isso, existe um *service worker*, que atua como um *proxy*, fazendo o controle do tráfego da aplicação, assim possibilitando o funcionamento do sistema mesmo sem conectividade com a internet. O sistema ainda faz uso da *IndexedDB*, uma API própria para armazenamento de dados no cliente, para guardar localmente o baralho atualmente em estudo e os *flashcards* que o compõe, assim como a classificação de cada *flashcard* no sistema Leitner. Esta arquitetura é exemplificada na “Figura 4 - Arquitetura do sistema”.

### 3.2.1 Requisitos funcionais

Nesta seção serão apresentados os requisitos funcionais, ou seja, os problemas e necessidades que são atendidos por funções do aplicativo ou serviços. O “Quadro 1 - Requisitos Funcionais” apresenta os requisitos funcionais esperados que o sistema forneça ao usuário.

**Quadro 1 - Requisitos Funcionais**

<b>Código</b>	<b>Descrição</b>
RF01	O sistema deve permitir que usuários se cadastrem
RF02	O sistema deve permitir que usuários cadastrados entrem no sistema
RF03	O sistema deve permitir que usuários logados obtenham uma listagem de seus baralhos
RF04	O sistema deve permitir que usuários logados criem um novo baralho
RF05	O sistema deve permitir que usuários logados obtenham uma listagem de cartas de um baralho selecionado
RF06	O sistema deve permitir que usuários logados criem uma nova carta para um baralho de sua autoria selecionado
RF07	O sistema deve permitir que usuários logados editem o conteúdo de uma carta para um baralho de sua autoria selecionado
RF08	O sistema deve permitir que usuários logados excluam uma carta de um baralho de sua autoria selecionado
RF09	O sistema deve permitir que usuários logados copiem uma carta de um baralho

	selecionado para outro baralho de sua autoria
RF10	O sistema deve permitir que usuários logados criem um novo baralho, duplicando um já existente, de sua autoria ou não
RF11	O sistema deve permitir que usuários logados excluam um baralho de sua autoria
RF12	O sistema deve permitir que usuários logados compartilhem um baralho de sua autoria
RF13	O sistema deve permitir que usuários logados se inscrevam no baralho de outro usuário
RF14	O sistema deve permitir que usuários logados configurem o tempo de espaçamento de repetição de cada baralho
RF15	O sistema deve permitir que usuários logados selecionem um baralho para estudos
RF16	O sistema deve notificar o usuário quando o tempo de espaçamento de repetição do baralho selecionado para estudos for atingido, para que o usuário possa iniciar uma rodada de estudos
RF17	O sistema deve permitir que um usuário logado inicie uma rodada de estudos de um baralho específico a qualquer momento
RF18	O sistema deve sortear uma carta, conforme a sua classificação e a lógica do método de Leitner, para apresentar ao usuário
RF19	O sistema deve reclassificar uma carta sorteada segundo a resposta do usuário, conforme a lógica do método de Leitner
RF20	O sistema deve atualizar a classificação das cartas no servidor, quando houver conectividade
RF21	O sistema deve guardar a classificação das cartas quando não houver conectividade, para posterior atualização no servidor
RF22	O sistema deve guardar o baralho selecionado para estudo, e a respectiva classificação de suas cartas, para uso caso não haja conectividade

Fonte: Elaborado pelo autor, 2021

### 3.2.2 Requisitos não funcionais

Os requisitos não funcionais, listados no “Quadro 2 - Requisitos não funcionais” descrevem qualidades globais do sistema, entre eles estão os requisitos que definem o aplicativo como um PWA.

Quadro 2 - Requisitos não funcionais

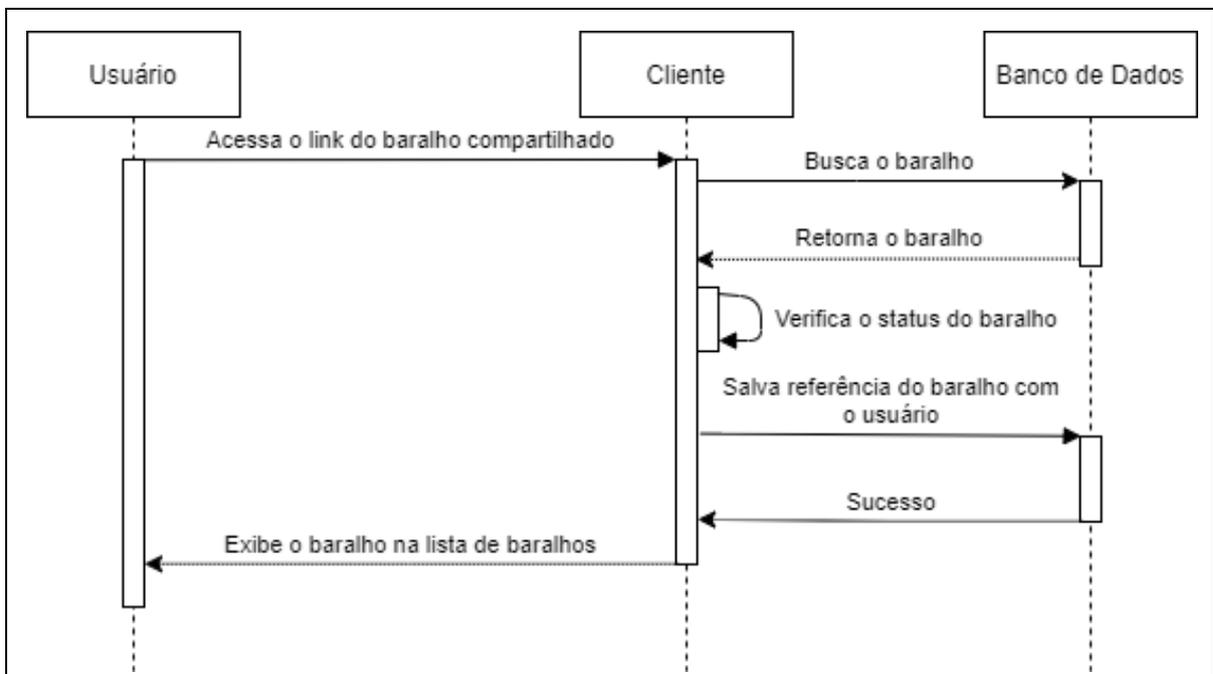
Código	Descrição
RNF01	O sistema deve apresentar design responsivo
RNF02	O sistema deve funcionar, após instalação, mesmo sem conectividade
RNF03	O sistema deve apresentar interações e navegação semelhantes a um app nativo
RNF04	O sistema deverá utilizar um <i>service worker</i> para a sua atualização
RNF05	O sistema deverá utilizar o protocolo HTTPS
RNF06	O sistema utilizará um manifesto para sua identificação como PWA
RNF07	O sistema utilizará recursos de reengajamento, como notificações <i>push</i>
RNF08	O sistema poderá ser instalado a partir do browser
RNF09	O sistema será acessível na web, por um link, para posterior instalação

Fonte: Elaborado pelo autor, 2021

### 3.2.3 Diagrama de sequência

Um diagrama de sequência “é uma figura que mostra os eventos que os atores externos geram, sua ordem e os eventos entre sistemas” (LARMAN, 2007, p.143). Na “Figura 5 - Diagrama de sequência de fluxo de inscrição”, um exemplo de como é o fluxo de inscrição de um usuário em um baralho disponibilizado por outro usuário, caso ocorra sem erros.

Figura 5 - Diagrama de sequência de fluxo de inscrição



Fonte: Elaborado pelo autor, 2021

Nesse diagrama o ator usuário acessa um link contendo o identificador de um baralho existente no sistema. Com isso, o cliente envia uma requisição para o servidor do banco de dados para que este retorne os dados do baralho pesquisado. Caso o baralho esteja liberado para compartilhamento, o servidor do banco de dados retorna os dados do baralho para o cliente. Com estes dados, o cliente verifica o status do baralho, que podem ser “público”, que se encontra ativo, ou “cópia”, que só permite a cópia do baralho. Na sequência com o resultado positivo, envia uma requisição para o servidor de banco de dados para que este persista a

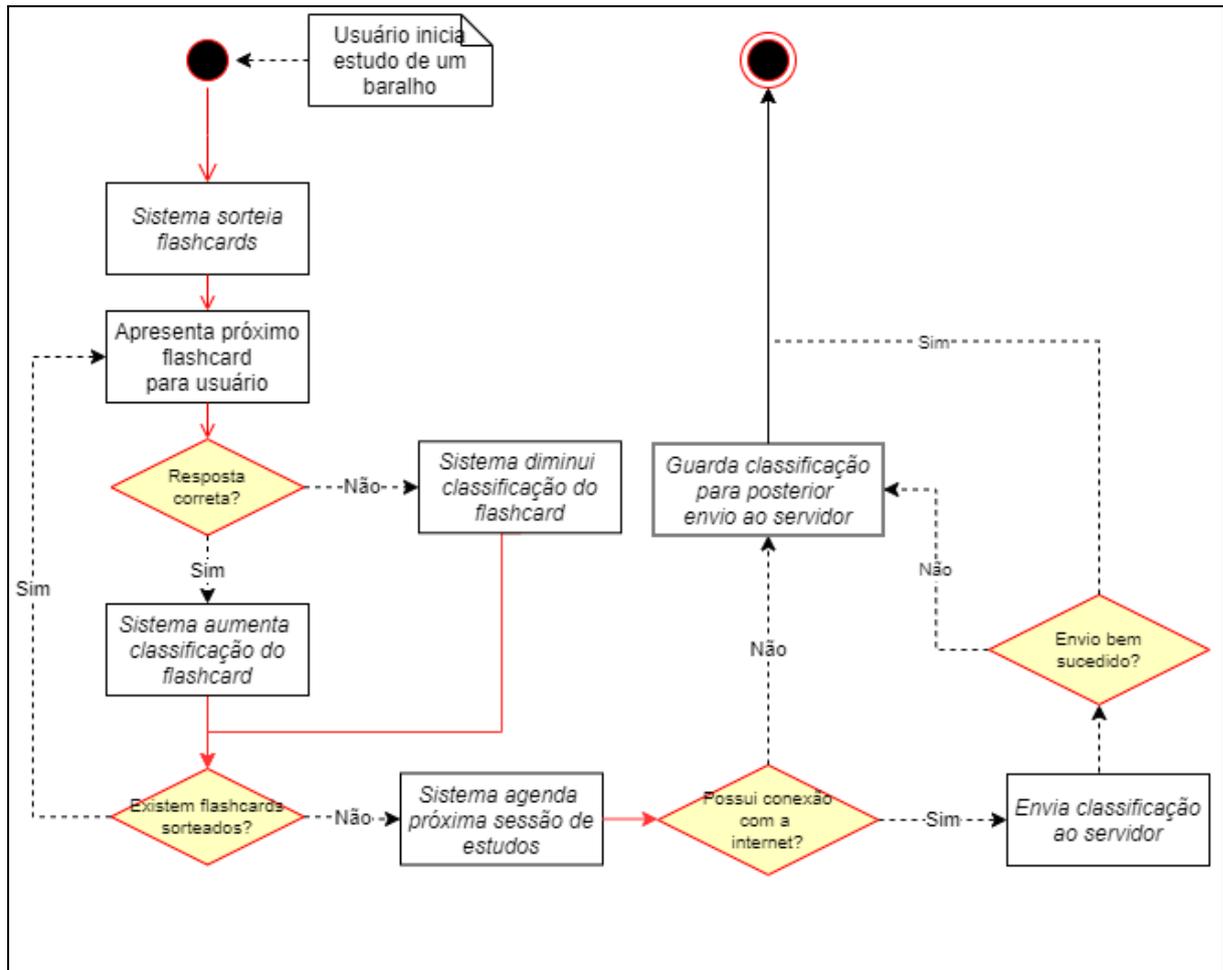
associação entre o usuário e o baralho. Após o sucesso da operação, o cliente exibe o baralho para o usuário, finalizando o fluxo.

#### **3.2.4 Diagrama de atividade**

O diagrama de atividade “é utilizado para representar os aspectos dinâmicos de um processo computacional” (FERREIRA e MARTINS, 2009, p.3). Seu objetivo é detalhar “o fluxo de controle e o fluxo de dados de uma determinada atividade, mostrando as ramificações de controle e as situações onde existem processamento paralelo” (FERREIRA e MARTINS, 2009, p.3).

O fluxo de cada rodada de estudos ocorre conforme ilustrado na “Figura 6 - Diagrama de atividade de fluxo de estudo” e descrito a seguir. Quando o usuário iniciar o estudo de um baralho, o sistema sorteia os *flashcards* a serem exibidos. Este sorteio é baseado na classificação do *flashcard*, conforme o sistema Leitner. Um a um os *flashcards* são apresentados ao usuário, esperando a resposta de acerto ou erro do usuário. Após, de acordo com a resposta, o sistema aumenta ou diminui a classificação de cada *flashcard*. A próxima sessão de estudos é agendada, conforme configuração de cada baralho. Em seguida, o sistema verifica se existe conexão com a internet. Caso exista, ele envia a classificação ao servidor, do contrário, o sistema guarda a classificação localmente, para envio futuro.

Figura 6 - Diagrama de atividade de fluxo de estudo



Fonte: Elaborado pelo autor, 2020

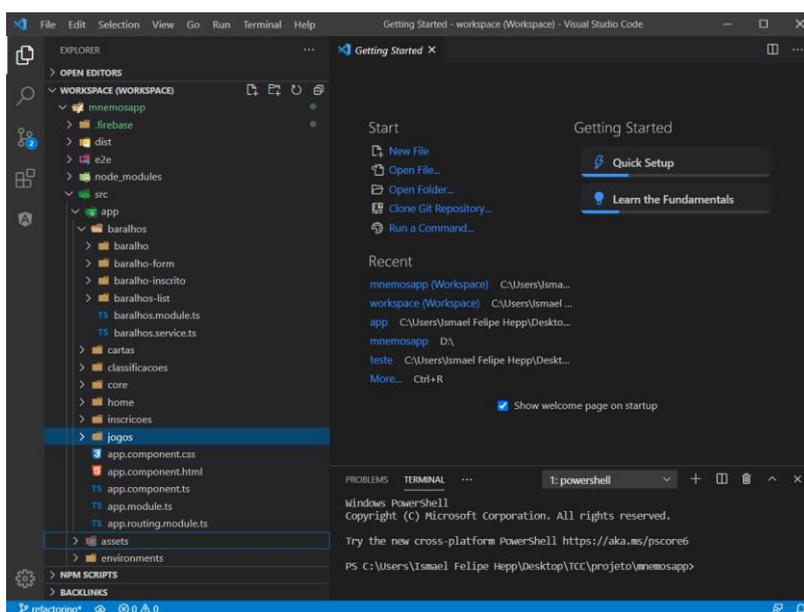
## 4 DESENVOLVIMENTO DO APLICATIVO

Este capítulo apresenta o processo do desenvolvimento do sistema, sua estrutura, os serviços que lhe dão suporte, e as ferramentas e tecnologias utilizadas pelo mesmo.

### 4.1 FERRAMENTAS

No desenvolvimento do sistema foi utilizado o Visual Studio Code, ilustrado na “Figura 7 - Tela inicial do Visual Studio Code”, como ambiente de desenvolvimento integrado (*integrated development environment*, IDE, no original em inglês). Um IDE é uma ferramenta que combina algumas ferramentas comuns de desenvolvimento em um mesmo programa, facilitando o processo. Alguns exemplos de ferramentas geralmente oferecidas em um IDE são o editor do código fonte, com funcionalidades como destaque de sintaxe e preenchimento automático conforme a linguagem utilizada; compilação de código em código binário; e *debugger*, ferramenta que, durante a execução do código, permite que o desenvolvedor acompanhe seu fluxo, os valores atribuídos às variáveis e retornos de funções, inspeção dos erros gerados durante a execução, entre outros.

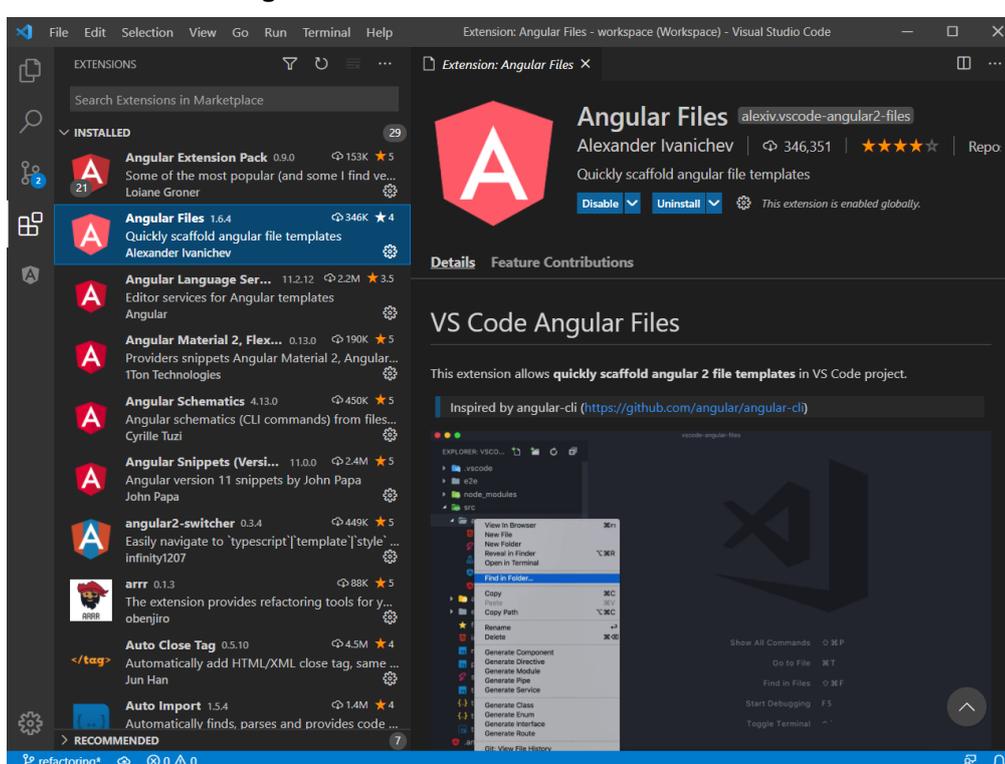
Figura 7 - Tela inicial do Visual Studio Code



Fonte: Elaborado pelo autor, 2021

O Visual Studio Code foi escolhido por ser uma ferramenta leve, gratuita, de código aberto e multiplataforma, que atende os requisitos básicos de um IDE e com a qual já se possuía experiência prévia de desenvolvimento. Outro motivo foi o suporte ao Typescript, linguagem utilizada no projeto, que é incluído por padrão na ferramenta (MICROSOFT, 2021). Além disso, ela possui uma vasta biblioteca de extensões, ilustrado na “Figura 8 - Extensões do Visual Studio Code”, que podem ser adicionadas ao projeto de maneira simples e que facilitam o desenvolvimento.

Figura 8 - Extensões do Visual Studio Code



Fonte: Elaborado pelo autor, 2021

Dentre as várias extensões utilizadas no projeto, destacamos as seguintes pelas suas funcionalidades relacionadas ao framework adotado:

- **Angular Files:** desenvolvida por Alexander Ivanichev, esta extensão adiciona funcionalidades ao IDE que possibilitam a rápida criação de componentes através de modelos de arquitetura do angular (scaffolding, no original);
- **Angular Language Service:** desenvolvida pela própria equipe responsável pelo Angular, a extensão adiciona ao IDE preenchimento de código automático relativo ao framework, documentação das funções nativas do

framework, mensagens de diagnóstico de performance, entre outras funcionalidades;

- **Angular Snippets (Version 11):** desenvolvida por John Papa, adiciona *snippets*, atalhos para estruturas de código recorrentes, relacionados ao *framework* Angular.

## 4.2 FRAMEWORK

Um *framework* é um conjunto de bibliotecas, funções, códigos, lógicas comuns em um desenvolvimento de software unidos em um mesmo pacote, seguindo um mesmo padrão de desenvolvimento. Ele serve para acelerar e simplificar o desenvolvimento de um software. Para este projeto, foi utilizado o Angular como *framework*.

Conforme documentado pelo Google (2021a), o Angular é uma plataforma de desenvolvimento que inclui, além de um *framework* baseado em componentes, uma coleção de bibliotecas integradas entre si que atendem a diversas variedades de funcionalidades necessárias comumente em aplicações e um conjunto de ferramentas que auxiliam no desenvolvimento e testes do software.

O Angular segue um padrão de desenvolvimento, adotado neste projeto. Este padrão possui algumas ideias principais, citadas a seguir:

- **Componentes:** São os blocos fundamentais que compõem a aplicação, cada um consiste em um modelo em HTML que possui as instruções de como o componente será apresentado, uma classe *Typescript* que define o comportamento do componente, um seletor CSS que define como o componente é utilizado em um *template* e um estilo em CSS, que define a aparência de seus elementos (GOOGLE, 2021a).
- **Templates:** São trechos de marcação HTML que definem a apresentação de um componente. No Angular, estes *templates* possuem sintaxe adicional que agregam diversas funcionalidades ao HTML, entre elas, por exemplo, a inserção de valores dinâmicos, a exibição condicional de certos elementos e a atualização da apresentação conforme a lógica do componente (GOOGLE, 2021a).
- **Diretivas:** São classes que adicionam ou modificam comportamentos de elementos já existentes. Existem em diferentes tipos como: componentes, já

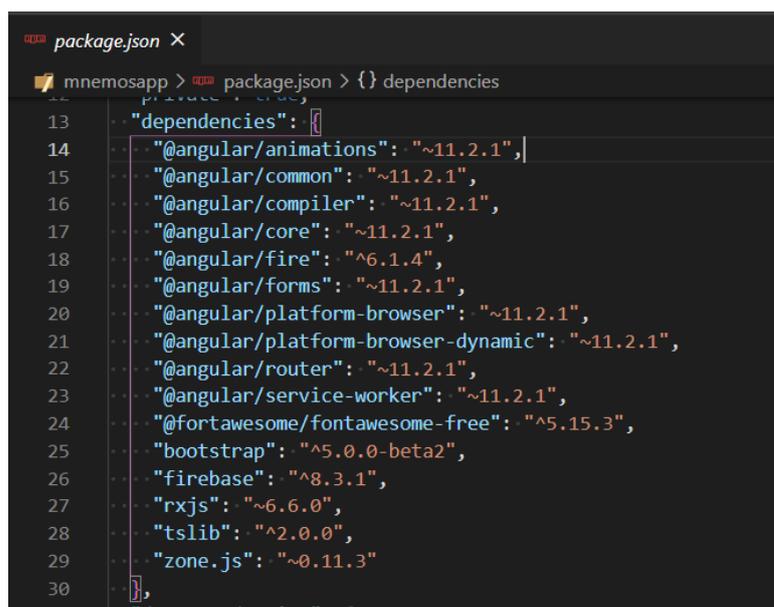
citados; diretivas de atributo, que mudam a aparência ou comportamento de um elemento, componente ou outra diretiva; e diretivas estruturais, que alteram o *layout* da aplicação ao adicionar ou remover elementos deste (GOOGLE, 2021a).

- **Injeção de dependência:** É um padrão utilizado pelo Angular que permite que se as dependências de uma classe do projeto sem precisar instanciá-las, o Angular faz isto automaticamente. As dependências são classes, serviços, objetos que são necessários para o funcionamento de outra classe. Para isto é necessário seguir o padrão utilizado pelo *framework*, fazendo a declaração destas dependências em cada classe que as utilizam (GOOGLE, 2021a).

### 4.3 BIBLIOTECAS

Um projeto Angular é modular, utilizando diversas bibliotecas em sua composição. Muitas delas são adicionadas por padrão ao projeto, em sua inicialização, por possuírem funções essenciais a execução de uma aplicação em Angular. Outras são adicionadas conforme a necessidade da aplicação. Na “Figura 9 - Bibliotecas do projeto”, as bibliotecas utilizadas no projeto são listadas.

Figura 9 - Bibliotecas do projeto



```
package.json X
mneiosapp > package.json > {} dependencies
13 "dependencies": {
14   "@angular/animations": "~11.2.1",
15   "@angular/common": "~11.2.1",
16   "@angular/compiler": "~11.2.1",
17   "@angular/core": "~11.2.1",
18   "@angular/fire": "^6.1.4",
19   "@angular/forms": "~11.2.1",
20   "@angular/platform-browser": "~11.2.1",
21   "@angular/platform-browser-dynamic": "~11.2.1",
22   "@angular/router": "~11.2.1",
23   "@angular/service-worker": "~11.2.1",
24   "@fortawesome/fontawesome-free": "^5.15.3",
25   "bootstrap": "^5.0.0-beta2",
26   "firebase": "^8.3.1",
27   "rxjs": "~6.6.0",
28   "tslib": "^2.0.0",
29   "zone.js": "~0.11.3"
30 },
31 "devDependencies": {
```

Fonte: Elaborado pelo autor, 2021

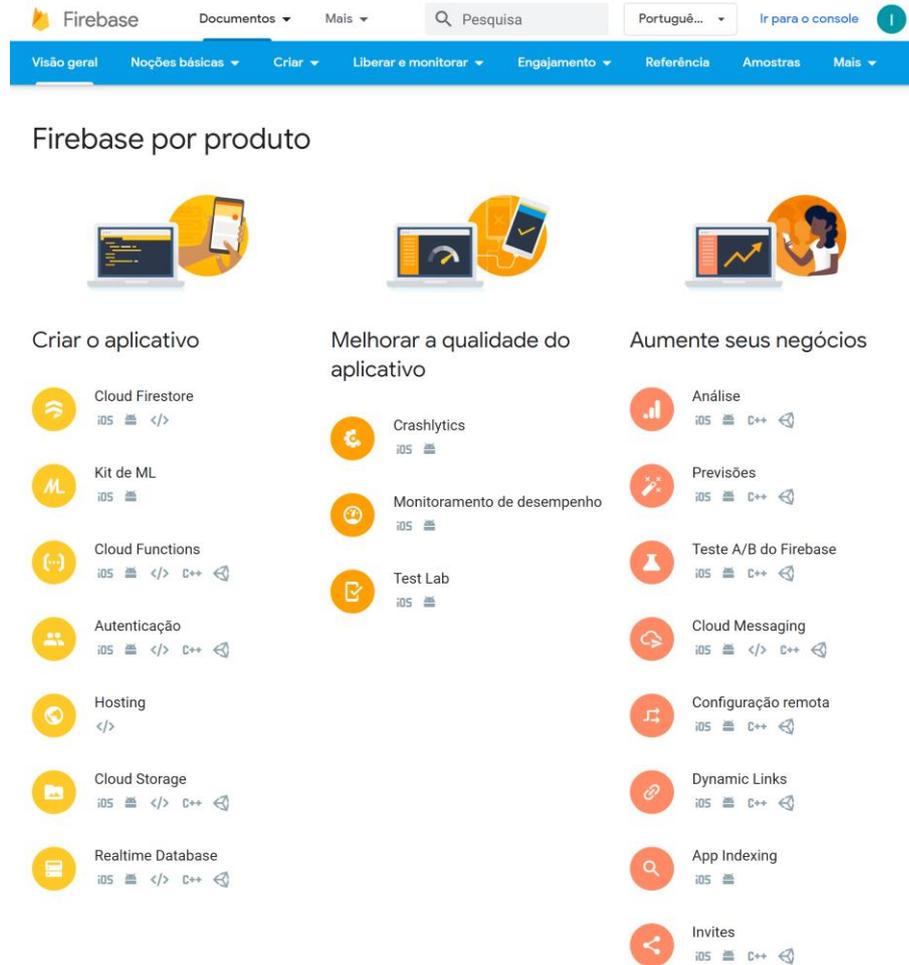
A seguir, serão comentadas as bibliotecas que foram adicionadas ao projeto para atender a fins específicos:

- **@angular/service-worker** - Biblioteca oficial do time do *Angular*, responsável por adicionar e configurar os arquivos necessários para que o *service worker* seja utilizado pela aplicação (GOOGLE, 2021b). Necessária para transformar a aplicação Angular em um PWA. Sua adição no projeto, por si só, já cria os arquivos e funcionalidades básicos de um PWA, mas a biblioteca conta com um arquivo de configuração próprio para o ajuste de cada parâmetro do PWA.
- **firebase** - Biblioteca oficial do time do Firebase. Possui funções específicas para o uso com os serviços da plataforma. Sua função é ligar a aplicação do usuário final com os serviços específicos utilizados pela mesma (GOOGLE, 2021c).
- **@angular/fire** - Apesar de não ser desenvolvida pelo time do Angular, esta biblioteca é indicada por eles para o uso do Firebase em uma aplicação Angular. Ela possui funções para o acesso aos serviços do Firebase, implementadas seguindo a lógica do framework Angular (ANGULAR, 2021).
- **bootstrap@next** - Biblioteca com a versão 5.0 do Bootstrap, um framework *front-end* voltado para agilizar e padronizar o desenvolvimento de *layouts* (BOOTSTRAP, 2021).

#### 4.4 SERVIÇOS UTILIZADOS

No aplicativo foram utilizados os serviços do Firebase como *backend*. O Firebase é um *backend as a service* (Baas) da Google, ou seja, uma plataforma que disponibiliza diversos serviços essenciais para o funcionamento de uma aplicação na web, conforme exemplificado na “Figura 10 - Serviços disponibilizados pelo Firebase”, para consumo direto, sem necessitar de um *backend* próprio da aplicação (GOOGLE, 2021d).

**Figura 10 - Serviços disponibilizados pelo Firebase**



**Fonte: Elaborado pelo autor, 2021**

Toda a estrutura do *backend*, como configuração do servidor, integração com banco de dados, autenticação, entre outros, são oferecidos como serviços prontos para integração com um aplicativo. Os serviços utilizados foram:

- **Firestore Authentication:** serviço utilizado para a autenticação dos usuários da aplicação. Possui integração com os outros serviços disponíveis na plataforma Firebase e oferece suporte à autenticação utilizando senhas, telefone, provedores de identidade como Google, Facebook e Twitter (GOOGLE, 2021d). No projeto, a fim de uma rápida implementação, só foi utilizada a autenticação a partir de uma conta da Google.
- **Cloud Firestore:** serviço que disponibiliza um banco de dados NoSQL, baseado em coleções e documentos (GOOGLE, 2021d). Utilizado no projeto para a persistência dos dados dos usuários, seus baralhos, cartas,

classificações e inscrições. Possui regras de segurança de escrita e leitura, podendo ser personalizada para cada recurso ou coleção de recursos.

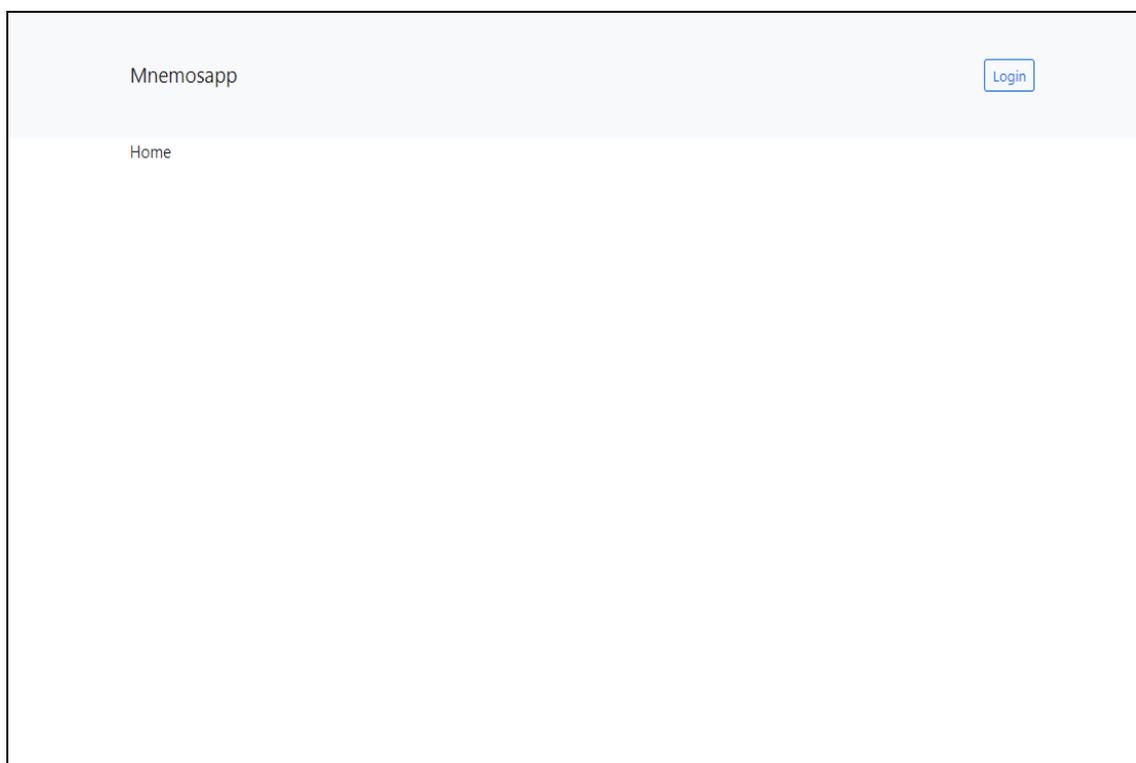
- **Firestore**: serviço de armazenamento de dados em nuvem, sua escolha, além da integração com os outros serviços, se deve ao fato de oferecer conexão em HTTPS (GOOGLE, 2021d), necessária para o uso de PWA. Além disso, oferece praticidade na implantação da aplicação, pois disponibiliza uma ferramenta de linha de comando, pela qual é possível implantar as alterações de código com poucos comandos.

## 5 RESULTADOS

Nesse capítulo será apresentado o resultado do desenvolvimento da aplicação. Serão apresentadas as funcionalidades presentes no aplicativo, com capturas de imagens das telas e do código do mesmo e a explicação das lógicas empregadas para a execução de suas funcionalidades.

Ao acessar a aplicação é exibida ao usuário a tela inicial do Mnemosapp, “Figura 11 - Tela Inicial do Mnemosapp”, onde, caso o usuário não esteja logado é possível fazer seu login através do botão "Login", localizado no cabeçalho da página.

**Figura 11 - Tela Inicial do Mnemosapp**



**Fonte: Elaborado pelo autor, 2021**

Nesta tela vários componentes da aplicação já são instanciados, para posterior uso pelos outros componentes. Primeiramente, o módulo principal da aplicação é instanciado, o “AppModule”. Este módulo importa os outros módulos utilizados pela aplicação, faz a configuração de alguns deles e, após isso, instancia o componente principal da aplicação, “AppComponent”. Na “Figura 12 – AppModule” uma captura do código, com os módulos do projeto.

Figura 12 - AppModule

```

TS app.module.ts 1, M X
mnmemosapp > src > app > TS app.module.ts > AppModule
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3  import { AppComponent } from './app.component';
4  import { AppRoutingModule } from './app.routing.module';
5  import { environment } from '../environments/environment';
6
7  import { AngularFireModule } from '@angular/fire';
8  import { AngularFireAuthModule } from '@angular/fire/auth';
9  import { AngularFireFirestoreModule } from '@angular/fire/firestore';
10
11 import { HomeModule } from './home/home.module';
12 import { CoreModule } from './core/core.module';
13 import { BaralhosModule } from './baralhos/baralhos.module';
14 import { CartasModule } from './cartas/cartas.module';
15 import { JogosModule } from './jogos/jogos.module';
16 import { InscricoesModule } from './inscricoes/inscricoes.module';
17 import { ServiceWorkerModule } from '@angular/service-worker';
18
19 @NgModule({
20   declarations: [
21     AppComponent
22   ],
23   imports: [
24     BrowserModule,
25     AngularFireModule.initializeApp(environment.firebaseConfig),
26     AngularFireAuthModule,
27     AngularFireFirestoreModule,
28     AppRoutingModule,
29     HomeModule,
30     CoreModule,
31     BaralhosModule,
32     CartasModule,
33     JogosModule,
34     InscricoesModule,
35     ServiceWorkerModule.register('ngsw-worker.js', {
36       enabled: environment.production,
37       registrationStrategy: 'registerImmediately'
38     })
39   ],
40   providers: [],
41   bootstrap: [AppComponent]
42 })
43 export class AppModule {}
44

```

Fonte: Elaborado pelo autor, 2021

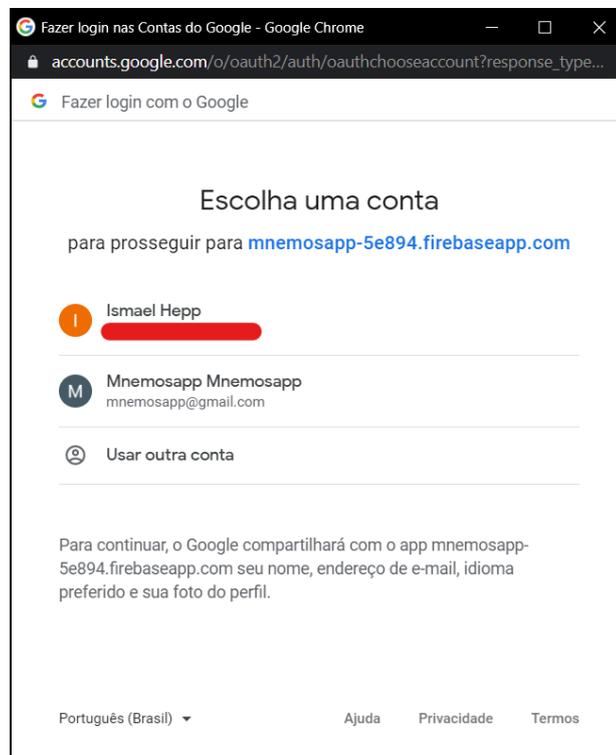
O *template* do “AppComponent” possui um elemento nativo do *framework* Angular, “*router-outlet*”, responsável por exibir um determinado componente de acordo com a rota acessada. A configuração da rota e o componente a ser exibido é feita no componente “AppRoutingModule”, que é instanciado pelo “AppModule”.

Outro elemento instanciado no “AppComponent” é o “*app-header*”, responsável por instanciar o “HeaderComponent”. Este componente é o cabeçalho da aplicação e está presente em todas as telas dela. Ao ser instanciado, é feita a injeção de todos os *services*, componentes responsáveis pela lógica de negócios da aplicação. São eles, “AuthService”, “BaralhosService”, “ClassificacoesService” e “InscricoesService”. Quando o usuário é logado no sistema estes componentes fazem uma requisição ao serviço de banco de dados para buscar os dados de cada componente referentes ao

usuário. O “HeaderComponent” só possui dois métodos autoexplicativos, *login* e *logout*.

Ao clicar no botão de *login* é apresentado ao usuário a *interface* de *login* utilizando o provedor de identidade do Google, ilustrado na “Figura 13 - Tela Autenticação do Mnemosapp”. Este processo é gerido pela biblioteca de autenticação do Firebase, o sistema apenas recebe os dados do usuário e, com eles, faz as requisições para obter os dados referentes aos baralhos, classificações e inscrições do usuário.

**Figura 13 - Tela Autenticação do Mnemosapp**



**Fonte:** Elaborado pelo autor, 2021

A “Figura 14 - Tela Inicial do Mnemosapp com usuário logado” exemplifica um usuário logado na tela inicial. O componente do cabeçalho passa a exibir os *links* que levam a página inicial, a página dos baralhos, o nome do usuário e o botão de *logout*.

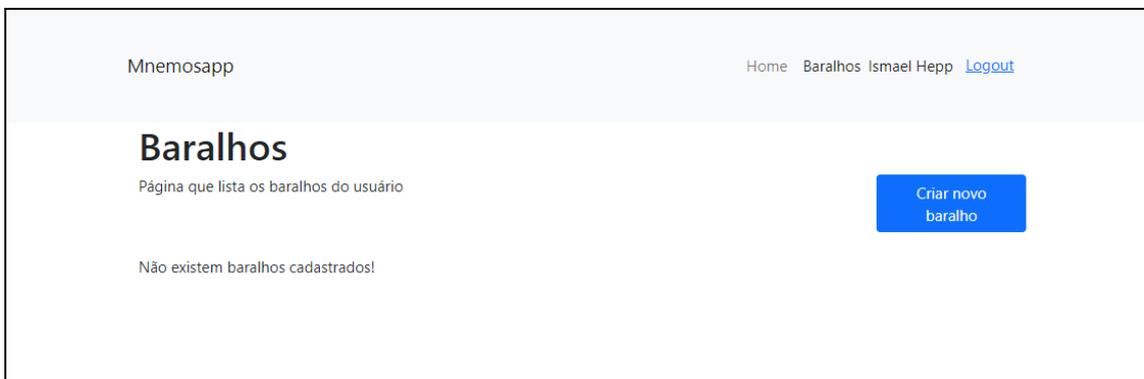
**Figura 14 - Tela Inicial do Mnemosapp com usuário logado**



**Fonte:** Elaborado pelo autor, 2021

Ao acessar o *link* “Baralhos”, o componente “BaralhoListComponent” é instanciado. Ele é o responsável por apresentar os baralhos criados e inscritos do usuário e, além disso, apresenta o botão “Criar novo baralho”, que é o responsável por acionar a função “criarBaralho” do componente. Na “Figura 15 - Tela de listagem de baralhos” temos um exemplo do visual do componente de um usuário sem baralhos criados nem inscritos.

**Figura 15 - Tela de listagem de baralhos**

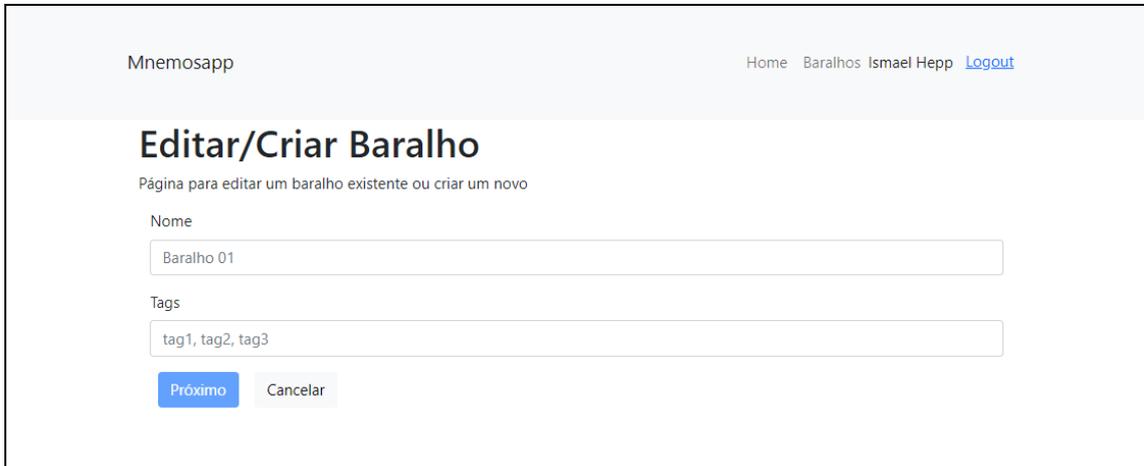


**Fonte:** Elaborado pelo autor, 2021

Quando o usuário pressiona o botão "Criar novo baralho" o componente “BaralhoFormComponent” é instanciado e exibido. Temos sua representação na “Figura 16 - Tela de criação de baralho”. Este componente é responsável por receber e fazer a validação dos dados do baralho do usuário. São eles o nome, *tags* e três cartas, formadas por suas respectivas perguntas e respostas. A qualquer momento é possível cancelar o procedimento, clicando no botão “Cancelar”, o que destrói a

instância do componente “BaralhoFormComponent” e volta a exibir o componente “BaralhoListComponent”.

**Figura 16 - Tela de criação de baralho**



Mnemosapp Home Baralhos Ismael Hepp Logout

## Editar/Criar Baralho

Página para editar um baralho existente ou criar um novo

Nome

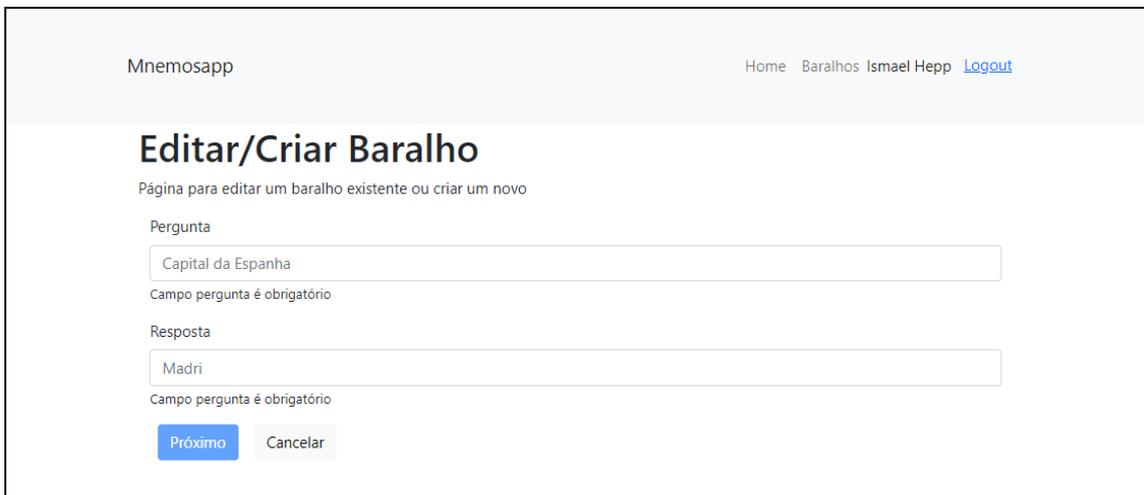
Tags

Próximo Cancelar

**Fonte: Elaborado pelo autor, 2021**

Na “Figura 17 - Tela de criação de baralho - criação de cartas” temos ilustrada a outra etapa de criação do baralho.

**Figura 17 - Tela de criação de baralho - criação de cartas**



Mnemosapp Home Baralhos Ismael Hepp Logout

## Editar/Criar Baralho

Página para editar um baralho existente ou criar um novo

Pergunta

Campo pergunta é obrigatório

Resposta

Campo pergunta é obrigatório

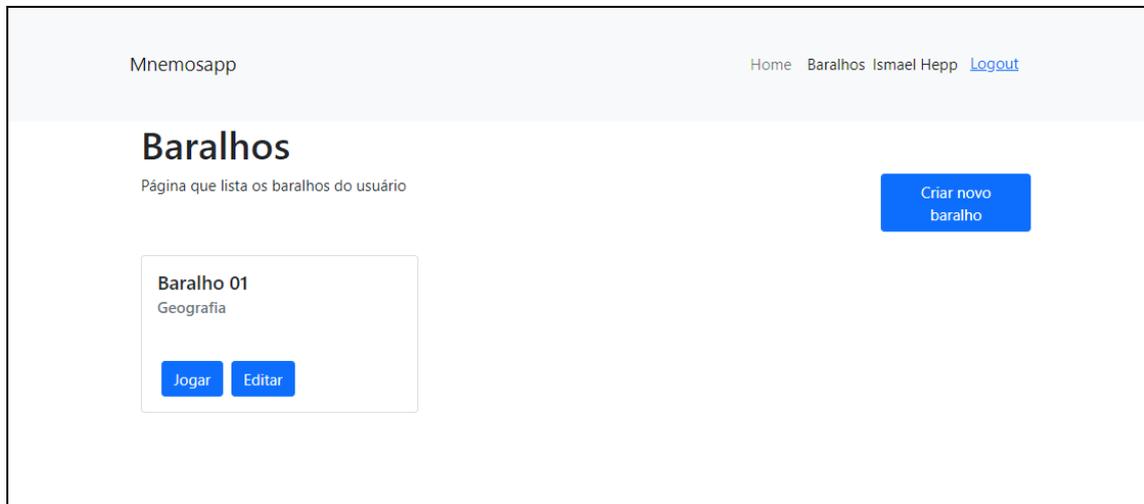
Próximo Cancelar

**Fonte: Elaborado pelo autor, 2021**

Ao validar os dados inseridos pelo usuário, o “BaralhoFormComponent” constrói um objeto Baralho com os dados recebidos e o passa para os serviços responsáveis pela persistência dos dados do baralho e da classificação do mesmo no banco de dados. Finalizada esta etapa, o componente “BaralhoFormComponent” tem sua instância destruída, e o sistema volta a exibir o componente

“BaralhoListComponent”, que exibe o baralho criado em sua listagem, como exemplificado na “Figura 18 - Tela de listagem de baralhos com novo baralho”.

**Figura 18 - Tela de listagem de baralhos com novo baralho**



**Fonte: Elaborado pelo autor, 2021**

Esta listagem dos baralhos do usuário é representada pelo componente “BaralhoListComponent”. Cada instância deste componente é responsável por exibir o nome e *tags* de um baralho específico e, também, expõe aos usuários os botões “Jogar” e “Editar”, que chamam funções de mesmo nome no componente.

O botão “Editar” exibe ao usuário o componente “BaralhoFormComponent” passando para ele o baralho associado ao “BaralhoListComponent”. Tal fato muda o comportamento do “BaralhoFormComponent” que, ao contrário do descrito anteriormente que guiava o usuário em um passo-a-passo para a criação de um novo baralho, passa a exibir os dados do baralho e lista as suas cartas, conforme “Figura 19 - Tela de edição de baralho”.

Figura 19 - Tela de edição de baralho

Mnemosapp Home Baralhos Ismael Hepp [Logout](#)

## Editar/Criar Baralho

Página para editar um baralho existente ou criar um novo

Nome

Tags

Compartilhar

Link para compartilhamento

[Salvar](#) [Cancelar](#) [Delete](#)

### Cartas

[Adicionar](#)

#	Pergunta	Resposta	Ações	
1	Capital da Argentina	Buenos Aires	<a href="#">Editar</a>	<a href="#">Deletar</a>
2	Capital do Brasil	Brasília	<a href="#">Editar</a>	<a href="#">Deletar</a>
3	Capital do Peru	Lima	<a href="#">Editar</a>	<a href="#">Deletar</a>

Fonte: Elaborado pelo autor, 2021

Além da possibilidade de edição dos campos nome e *tags* do baralho, também é apresentado o campo “Compartilhar”. Ao ser selecionado, e após salvar a alteração, uma inscrição é criada e persistida no banco de dados, liberando o baralho para compartilhamento e exibindo o *link* para tal, como demonstrado na “Figura 20 - Tela de edição - baralho compartilhado”.

Figura 20 - Tela de edição - baralho compartilhado

Mnemosapp Home Baralhos Ismael Hepp [Logout](#)

## Editar/Criar Baralho

Página para editar um baralho existente ou criar um novo

Nome  
Baralho 01

Tags  
Geografia

Compartilhar

Link para compartilhamento  
<http://localhost:4200/inscrever/LAWS9mGi1e44sQuEjOZL>

[Salvar](#) [Cancelar](#) [Delete](#)

### Cartas

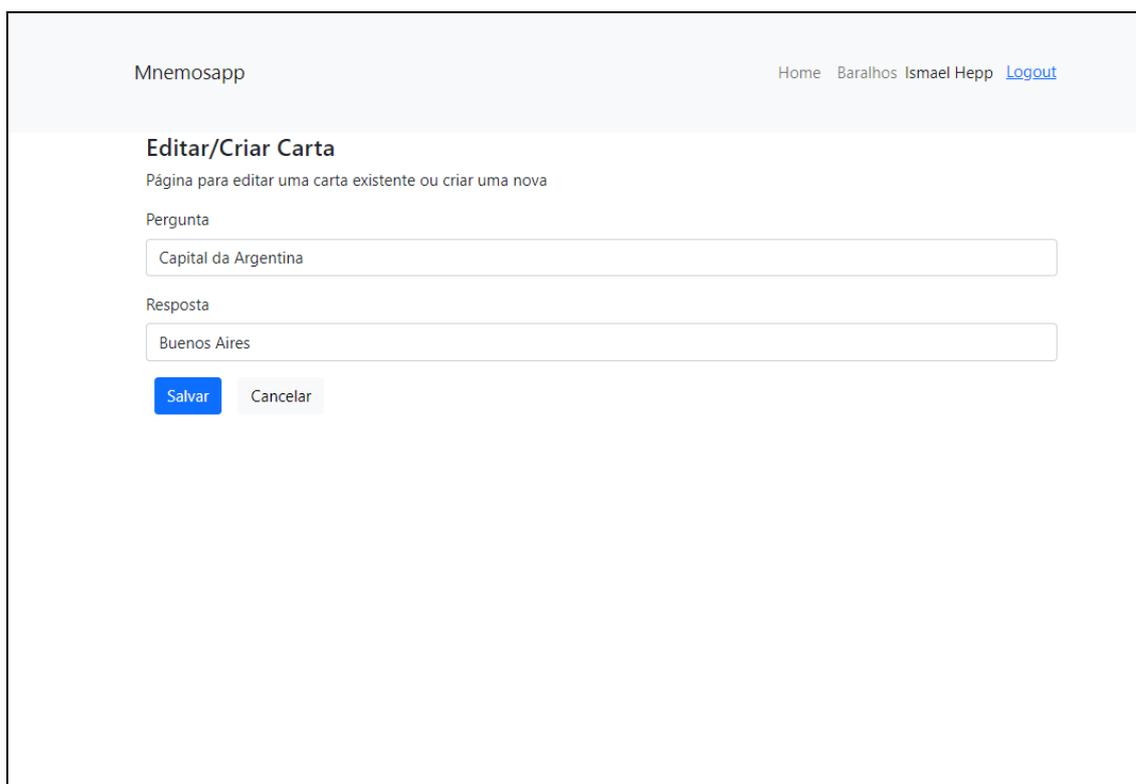
[Adicionar](#)

#	Pergunta	Resposta	Ações	
1	Capital da Argentina	Buenos Aires	<a href="#">Editar</a>	<a href="#">Deletar</a>
2	Capital do Brasil	Brasília	<a href="#">Editar</a>	<a href="#">Deletar</a>
3	Capital do Peru	Lima	<a href="#">Editar</a>	<a href="#">Deletar</a>

Fonte: Elaborado pelo autor, 2021

Ao acessar este *link* um usuário logado no sistema se inscreve no baralho compartilhado e, com isso, pode jogá-lo como se fosse seu.

Na listagem de cartas o usuário tem a opção de adicionar uma nova carta, deletar ou editar uma carta existente, como mostrado na “Figura 21 - Tela de criação e edição de carta”. Todas as opções atualizam a classificação do baralho de acordo, adicionando uma carta nova no bloco correto e removendo uma carta deletada de seu respectivo bloco.

**Figura 21 - Tela de criação e edição de carta**

Mnemosapp Home Baralhos Ismael Hepp [Logout](#)

### Editar/Criar Carta

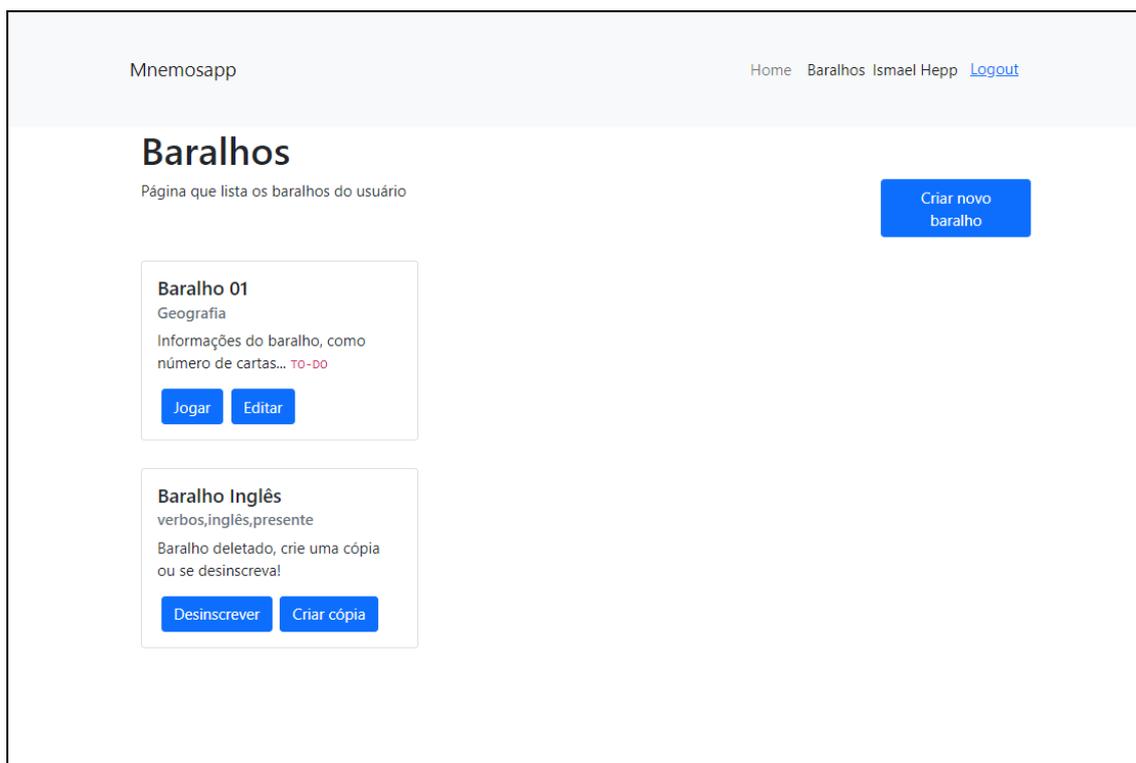
Página para editar uma carta existente ou criar uma nova

Pergunta

Resposta

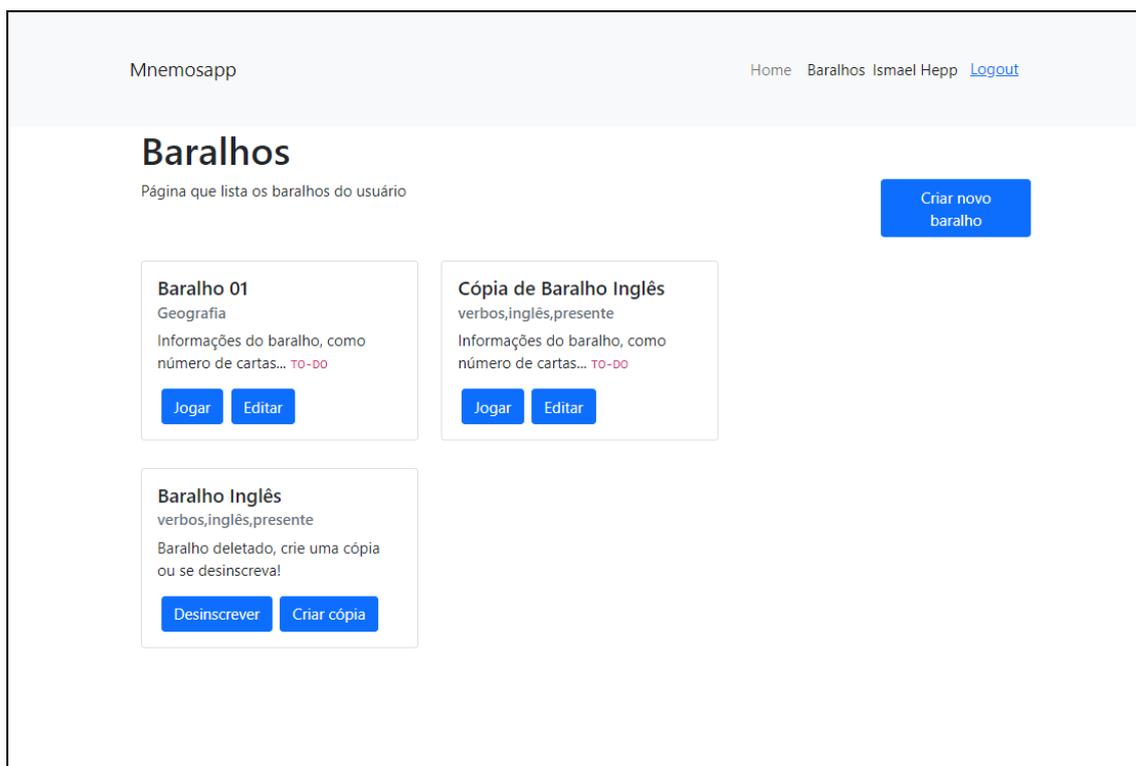
Fonte: Elaborado pelo autor, 2021

Por fim, é possível deletar o baralho através deste componente. Ao fazê-lo a classificação do usuário relacionada ao baralho também é excluída. Caso o baralho esteja liberado para compartilhamento, e existam outros usuários inscritos nele, o baralho não será excluído, ele apenas não será mais listado como disponível para o usuário que o criou. Neste caso, os usuários inscritos receberão a opção de fazer uma cópia própria do baralho para si ou remover a inscrição do baralho. Somente após todos os usuários inscritos escolherem uma das opções é que o baralho será realmente excluído. Na "Figura 22 - Tela de listagem de baralhos com baralho inscrito" temos um exemplo de baralho compartilhado em que o autor do baralho o excluiu.

**Figura 22 - Tela de listagem de baralhos com baralho inscrito**

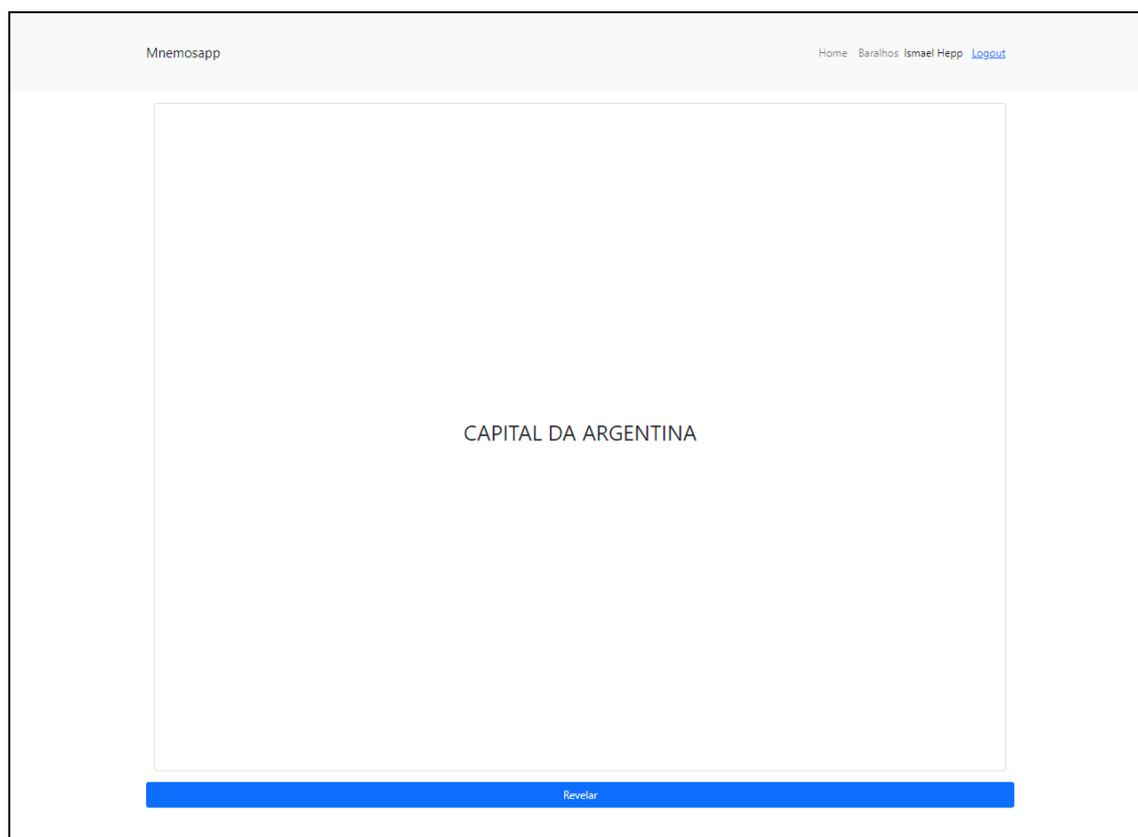
Fonte: Elaborado pelo autor, 2021

Na “Figura 23 - Tela de listagem de baralhos com cópia de baralho inscrito” temos um exemplo em que o usuário inscrito em um baralho que foi deletado usou a opção de criar uma cópia deste baralho para si.

**Figura 23 - Tela de listagem de baralhos com cópia de baralho inscrito**

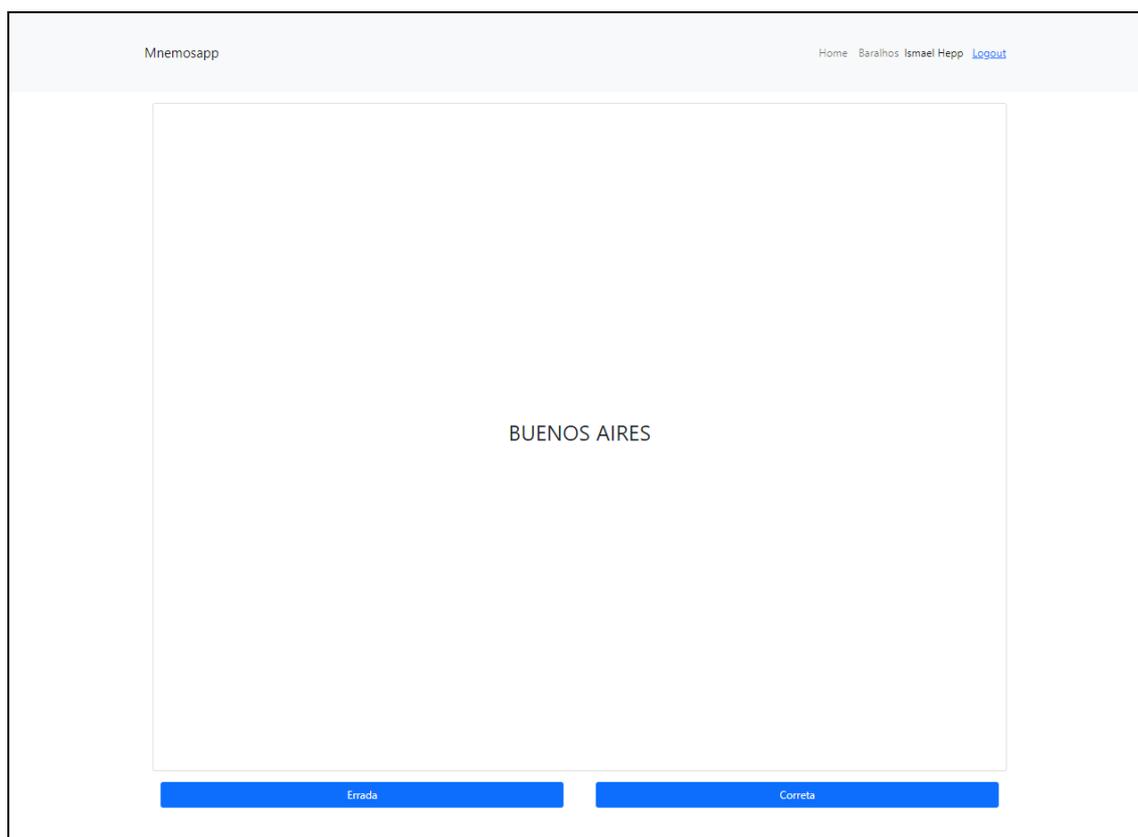
Fonte: Elaborado pelo autor, 2021

Por fim, a última funcionalidade oferecida pelo sistema é o da rodada de estudos. Ela é ativada para um específico baralho ao se pressionar o botão "Jogar" no mesmo. Após isto é instanciado o componente "JogoComponent", com os dados do baralho selecionado. Este componente faz o sorteio da carta a ser exibida, de acordo com a classificação de cada carta, e instancia o componente "CartaComponent" para exibi-la, conforme "Figura 24 - Tela da carta".

**Figura 24 - Tela da carta**

**Fonte: Elaborado pelo autor, 2021**

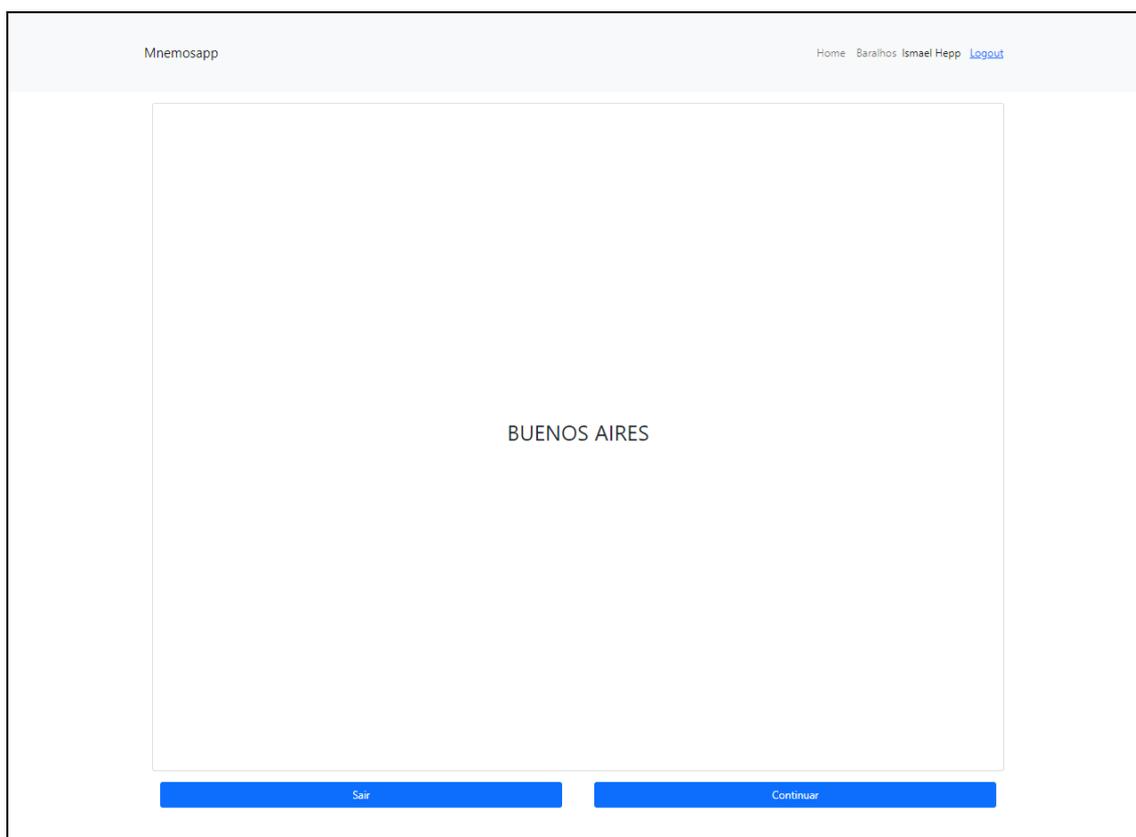
Em seguida o usuário decide quando revelar a resposta, pressionando o botão “Revelar”, o que exibe a resposta e dois novos botões, “Errada” e “Correta”, conforme “Figura 25 - Tela da carta com resposta revelada”.

**Figura 25 - Tela da carta com resposta revelada**

**Fonte: Elaborado pelo autor, 2021**

Dependendo da escolha do usuário a classificação da carta mudará. Caso o usuário marque sua resposta como certa, a carta subirá de classificação, que vai se 5 à 1, sendo 1 a mais alta e 3 a padrão para novas cartas. Isto influencia na lógica de sorteio, diminuindo a frequência com que ela será sorteada nas próximas rodadas de estudo. Caso o usuário marque como incorreta, o contrário é feito. Estas alterações são persistidas pelo “JogoComponent”, utilizando o *service* responsável pelo acesso aos dados das classificações, assim que o usuário defina sua resposta como correta ou incorreta.

Finalmente, conforme “Figura 26 - Fim de rodada”, ao usuário são apresentadas duas opções, encerrar a rodada de estudos através do botão "Sair", que o direciona à listagem de baralhos, ou continuar a rodada com o botão "Continuar", que repetirá o processo, sorteando uma nova carta com base nas classificações das cartas do baralho.

**Figura 26 - Fim de rodada**

**Fonte: Elaborado pelo autor, 2021**

Os testes de fluxo foram feitos no browser Chrome, no seu modo padrão e na emulação de dispositivos móveis. Os fluxos testados foram a autenticação de usuário; a criação, edição e exclusão de baralho de autoria do usuário e inscrito; a criação, edição e exclusão de cartas de autoria do usuário e de baralho inscrito; o compartilhamento, cópia, inscrição e cancelamento de inscrição de baralhos; a rodada de estudos e a reclassificação das cartas conforme o resultado da rodada, para baralhos de autoria do usuário como inscritos; e a instalação, desinstalação do aplicativo em um computador *desktop* e um celular *Android*. Todos os fluxos citados foram testados novamente em modo *off-line*.

Também foi utilizada a ferramenta *Lighthouse* do Chrome, que inspeciona as páginas da aplicação, checando sua performance, acessibilidade, se ela se enquadra como um aplicativo da web progressivo, entre outros. O aplicativo passou na auditoria da ferramenta.

## 6 CONCLUSÃO

O presente estudo teve como objetivo desenvolver um aplicativo da *web* progressivo de auxílio ao aprendizado, baseado na técnica de repetição espaçada com o uso de *flashcards*. Esta ferramenta foi projetada com o intuito de oferecer um objeto digital de aprendizagem de uso simples, tema livre e de fácil compartilhamento, para que professores pudessem criar um baralho e disponibilizar para estudo, objetivo atingido.

O levantamento de requisitos foi feito através de revisão das técnicas que embasam sua lógica, tal como de aplicativos semelhantes. De acordo com a revisão, foi feita a modelagem de dados e pesquisa de ferramentas de desenvolvimento.

O aplicativo foi desenvolvido utilizando o *framework* Angular, da Google, próprio para aplicações do gênero. Sua combinação com a metodologia PWA permitiu criar um aplicativo completo e mais leve, ao se comparar com aplicativos nativos.

O tempo disponível para o desenvolvimento foi a maior dificuldade encontrada durante o trabalho. Existiam planos de disponibilizar o aplicativo para uso, coletar as opiniões de seus usuários e fazer uma análise das mesmas, mas com o tempo disponível foi necessário se concentrar em seu desenvolvimento e em disponibilizar as principais funções da aplicação.

O aprendizado de um novo *framework*, com padrões próprios, também foi uma dificuldade, embora menor. Assim que a curva de aprendizado foi superada o desenvolvimento progrediu de maneira simples.

A utilização de PWA foi crucial no desenvolvimento do aplicativo pretendido, mas como se trata de uma metodologia ainda recente algumas funcionalidades ainda estão sendo desenvolvidas. Uma delas, que poderia ser utilizada no sistema, é a de tarefas programadas, que realiza uma tarefa em um horário pré-programado.

Os próximos passos para trabalhos futuros são a melhoria da *interface*, classificação de baralhos conforme suas *tags*, a automatização das rodadas de estudo através de tarefas programadas, e a disponibilização para uso e análise das opiniões de seus usuários.

## REFERÊNCIAS

ANGULAR. **AngularFire**. 2021. Disponível em: <<https://firebaseopensource.com/projects/angular/angularfire2/>>. Acesso em: 01 mai. 2021.

ABED, Associação Brasileira de Educação a Distância. **Censo EAD.BR 2018 Relatório analítico da aprendizagem a distância no Brasil**. Disponível em: <[http://abed.org.br/arquivos/CENSO\\_DIGITAL\\_EAD\\_2018\\_PORTUGUES.pdf](http://abed.org.br/arquivos/CENSO_DIGITAL_EAD_2018_PORTUGUES.pdf)>. Acesso em: 01 mai. 2021.

ASSUMPÇÃO, C. M. **Quais são os recursos de EAD que estão despontando no horizonte**. ABED-Associação Brasileira de Educação a Distância. Censo EAD. BR, 2018.

AZEVEDO, Lucas Wendell Barbosa. **My Flashcards: aplicativo multiplataforma para auxílio nos estudos utilizando flashcards**. 2017. Disponível em: <[http://www.repositorio.ufc.br/bitstream/riufc/29561/1/2017\\_tcc\\_lwbazevedo.pdf](http://www.repositorio.ufc.br/bitstream/riufc/29561/1/2017_tcc_lwbazevedo.pdf)>. Acesso em: 01 mai. 2021.

BOOTSTRAP. **Introduction**. 2021. Disponível em: <<https://getbootstrap.com/docs/5.0/getting-started/introduction/>>. Acesso em: 01 mai. 2021.

CAMARGO, Suélen Antunes. **Aplicativo multiplataforma para aprendizado de Hangul utilizando flashcards**. 2017. Disponível em: <<https://painel.passofundo.ifsul.edu.br/uploads/arq/201804121728031081110908.pdf>>. Acesso em: 01 mai. 2021.

**CURVA DO ESQUECIMENTO**. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2020. Disponível em: <[https://pt.wikipedia.org/w/index.php?title=Curva\\_do\\_esquecimento&oldid=59670949](https://pt.wikipedia.org/w/index.php?title=Curva_do_esquecimento&oldid=59670949)>. Acesso em: 01 mai. 2021.

FERREIRA, Jeferson; MARTINS, Eliane. **Análise de Fluxo de Controle e Dados a partir do Diagrama de Atividades da UML 2.0**. 2009.

FOLEY, Mary Jo. **Microsoft is automatically installing Office PWAs for some Windows 10 users: Report**. ZDNet. 14 out. 2020. Disponível em: <<https://www.zdnet.com/article/microsoft-is-automatically-installing-office-pwas-for-some-windows-10-users-report/>>. Acesso em: 01 mai. 2021.

GAUNT, M. **Introdução aos service workers**. 2020. Disponível em: <<https://developers.google.com/web/fundamentals/primers/service-workers>>. Acesso em: 01 mai. 2021.

GOOGLE. **Angular**. 2021a. Disponível em: <<https://angular.io/docs>>. Acesso em: 01 mai. 2021.

GOOGLE. **Angular service worker introduction**. 2021b. Disponível em: <<https://angular.io/guide/service-worker-intro>>. Acesso em: 01 mai. 2021.

- GOOGLE. **Firestore JavaScript SDK Reference**. 2021c. Disponível em: <<https://firebase.google.com/docs/web/setup>>. Acesso em: 01 mai. 2021.
- GOOGLE. **Documentação Firebase**. 2021d. Disponível em: <<https://firebase.google.com/docs?hl=pt-br>>. Acesso em: 01 mai. 2021.
- LARMAN, C. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos**. Porto Alegre: Bookman; 2007.
- MICROSOFT. **Documentation for Visual Studio Code**. 2021. Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 01 mai. 2021.
- MOGNON, Mateus. **Google substitui apps do Android por PWAs no Chrome OS**. Tecmundo. 14 abr. 2020. Disponível em: <<https://www.tecmundo.com.br/software/152045-google-substitui-apps-android-pwa-chrome.htm>>. Acesso em: 01 mai. 2021.
- ROCHA, Alberto. **Google e Microsoft se unem para desenvolver mais apps da web na Play Store**. Canaltech. 12 jul. 2020. Disponível em: <<https://canaltech.com.br/android/google-e-microsoft-se-unem-para-desenvolver-mais-apps-da-web-na-play-store-167926/>>. Acesso em: 01 mai. 2021.
- RUSSEL, A. **Progressive Web Apps Escaping Tabs Without Losing Our Soul**. 2015. Disponível em: <<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>>. Acesso em: 01 mai. 2021.
- SILVA, Diogo Correia Araujo. **Flashcards digitais - técnica de repetição espaçada aplicada ao apoio na memorização do conteúdo estudado**. Revista Gestão Universitária, p. 1-10, 2015. Disponível em: <<http://gestaouniversitaria.com.br/artigos/flashcards-digitais-tecnica-de-repeticao-espacada-aplicada-ao-apoio-na-memorizacao-do-conteudo-estudado>>. Acesso em: 01 mai. 2021.
- WHATWG, Web Hypertext Application Technology Working Group et al. **HTML Living Standard**. 2020. Disponível em: <<https://html.spec.whatwg.org/multipage/workers.html>>. Acesso em: 01 mai. 2021.